

**UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET**

Goran S. Savić

**HARDVERSKA REALIZACIJA BRZOG
DEKODERA KOMPRIMOVANE SLIKE
SA MINIMALNIM ZAHTEVANIM
RESURSIMA**

DOKTORSKA DISERTACIJA

Beograd, 2017.

**UNIVERSITY OF BELGRADE
SCHOOL OF ELECTRICAL ENGINEERING**

Goran S. Savić

**HARDWARE REALIZATION OF THE
FAST DECODER OF COMPRESSED
IMAGE WITH MINIMUM REQUIRED
RESOURCES**

DOCTORAL DISSERTATION

Belgrade, 2017.

Podaci o mentoru i članovima komisije

Mentor:

dr Milan Prokin, redovni profesor, Univerzitet u Beogradu, Elektrotehnički fakultet

Članovi komisije:

dr Milan Prokin, redovni profesor, Univerzitet u Beogradu, Elektrotehnički fakultet

dr Jelena Popović Božović, docent, Univerzitet u Beogradu, Elektrotehnički fakultet

dr Miroslav Lutovac, redovni profesor, Univerzitet Singidunum, Tehnički fakultet

dr Irini Reljin, redovni profesor, Univerzitet u Beogradu, Elektrotehnički fakultet

dr Ivan Popović, docent, Univerzitet u Beogradu, Elektrotehnički fakultet

dr Milan Ponjavić, vanredni profesor, Univerzitet u Beogradu, Elektrotehnički fakultet

Datum odbrane:

Hardverska realizacija brzog dekodera komprimovane slike sa minimalnim zahtevanim resursima

Rezime doktorske disertacije

U ovoj doktorskoj disertaciji je opisana hardverska realizacija novog brzog hardverskog dekodera komprimovane slike sa optimizovanim utroškom logičkih i memorijskih resursa, kao i minimizovanim kašnjenjem koje produkuje sistem za kompresiju i dekompresiju digitalne slike, koga čine već razvijeni hardverski enkoder i brzi hardverski dekoder opisan u ovoj disertaciji. Realizacija brzog hardverskog dekodera je verifikovana implementacijom unutar jednog programabilnog logičkog kola nižeg cenovnog razreda, bez korišćenja bilo kakve spoljašnje memorije. Svaki od blokova koji sačinjavaju dekoder za dekompresiju slike (entropijski dekoder, dekoderski procenjivač verovatnoće, dekvantizator, inverzni podopsežni transformator) je zasebno realizovan sa ciljem postizanja što veće uštede u utrošku logičkih i memorijskih resursa, ali je istovremeno vođeno računa i o optimizaciji utroška resursa i na nivou dekodera kao celine.

U disertaciji je predložena nova, hardverski efikasna, realizacija entropijskog dekodera, koja omogućava smanjenje utrošenih logičkih resursa u odnosu na realizacije entropijskog dekodera prema postojećem stanju tehnike, kao i nova, hardverski efikasna, realizacija dekoderskog procenjivača verovatnoće, sa optimizovanim utroškom memorijskih i logičkih resursa. U ovoj disertaciji je opisana i nova realizacija inverznog podopsežnog transformatora koji vrši dvodimenzionalnu inverznu diskretnu *wavelet* transformaciju (DWT) sa 5/3 filtrom koristeći pri tome 20% manje memorijskih resursa, kao i manju količinu logičkih resursa, u poređenju sa najboljom do sada

objavljenom realizacijom u literaturi. U disertaciji je predložena i nova, vremenski nestacionarna realizacija jednodimenzionalnog inverznog DWT sa 5/3 filtrom, efikasnija od bilo koje druge do sada objavljene realizacije u pogledu utroška logičkih i memorijskih resursa, maksimalne radne frekvencije i ukupne disipacije.

Ključne reči: dekompresija slike, hardverska realizacija, minimalni utrošeni resursi

Naučna oblast: tehničke nauke, elektrotehnika

Uža naučna oblast: elektronika

UDK broj: 621.3

Hardware Realization of the Fast Decoder of Compressed Image with Minimum Required Resources

Abstract of the Doctoral Dissertation

Hardware realization of a novel fast decoder of compressed image with an optimized utilization of logic and memory resources and minimized delay introduced by a system which consists of an encoder and the novel fast decoder, is described in this doctoral dissertation. Hardware realization of the novel fast decoder has been verified by implementation within a low cost programmable logic device, without any external memory components. Each block contained by the fast decoder of compressed image (entropy decoder, decoder probability estimator, dequantizer and inverse subband transformer) has been separately designed in order to increase the savings of logic and memory resources, while simultaneously optimizing the whole decoder as well.

In this doctoral dissertation, the novel efficient hardware realization of entropy decoder, which uses less logic resources than any other state-of-the-art realization, as well as the novel efficient hardware realization of decoder probability estimator with optimized use of memory and logic resources, has been proposed. Also, in this doctoral dissertation is described the novel hardware realization of the inverse subband transformer which is responsible for two-dimensional inverse discrete wavelet transform (DWT) with $5/3$ filter, using 20% less memory resources and using less logic resources than any other realization published so far in the literature. Finally, the novel non-stationary hardware realization of one-dimensional inverse DWT with $5/3$ filter, which is more efficient than any other reported realization in terms of amount of used

logic and memory resources, maximum operating frequency and total power dissipation, has been also presented in this doctoral dissertation.

Key words: image decompression, hardware realization, minimum required resources

Scientific area: technical sciences, electrical engineering

Specific scientific area: electronics

UDK number: 621.3

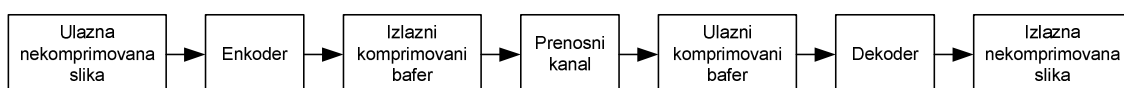
SADRŽAJ

1. Uvod.....	1
2. Entropijski dekodeer i dekoderski procenjivač verovatnoće.....	5
2.1. Uvod.....	5
2.2. Algoritam rada entropijskog dekodera i dekoderskog procenjivača verovatnoće.....	6
2.3. Hardverska realizacija entropijskog dekodera i dekoderskog procenjivača verovatnoće.....	30
3. Dekvantizator.....	39
3.1. Uvod.....	39
3.2. Hardverska realizacija dekvantizatora.....	42
4. Inverzni podopsežni transformator.....	45
4.1. Uvod.....	45
4.2. Hardverska realizacija jednodimenzionalnog inverznog DWT 5/3 filtra.....	51
4.2.1. Poređenja performansi različitih hardverskih realizacija jednodimenzionalnog inverznog DWT 5/3 filtra.....	61
4.2.2. Poređenja rezultata sinteze različitih hardverskih realizacija jednodimenzionalnog inverznog DWT 5/3 filtra.....	64
4.3. Hardverska realizacija dvodimenzionalnog inverznog DWT 5/3 filtra.....	67
4.3.1. Poređenja performansi različitih hardverskih realizacija dvodimenzionalnog inverznog DWT 5/3 filtra.....	84
4.3.2. Poređenja rezultata sinteze različitih hardverskih realizacija dvodimenzionalnog inverznog DWT 5/3 filtra.....	89
4.4. Blok šema najvišeg hijerarhijskog nivoa inverznog podopsežnog transformatora	91
5. Realizacija brzog dekodera za jednu ravan boje.....	94
6. Realizacija i implementacija brzog dekodera sa minimalnim utrošenim resursima za sliku u boji.....	99
7. Zaključak.....	110

Literatura.....	112
Biografija autora.....	120

1. Uvod

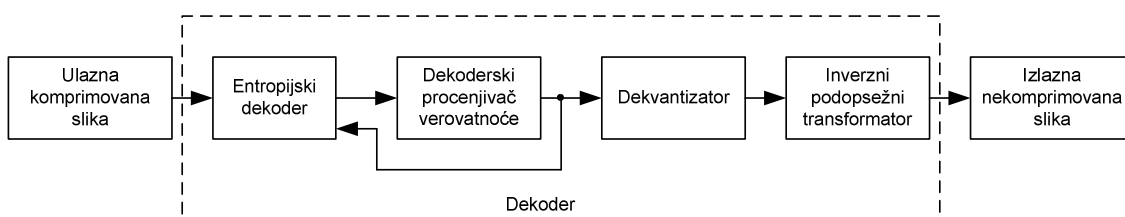
Razvoj novih i unapređenje postojećih tehnika za kompresiju i dekompresiju digitalne slike (nepokretne i pokretne) su danas veoma aktuelni. Postoji stalna potreba za poboljšanjem kvaliteta i rezolucije digitalne slike, potreba za povećanjem *frame rate*-a (broja slika u sekundi koje čine video sekvencu) i potreba za povećanjem dužine trajanja video zapisa. Sve to za posledicu ima povećanje utroška logičkih i memorijskih resursa potrebnih za skladištenje digitalne slike, povećanje utroška logičkih, memorijskih i vremenskih resursa potrebnih za obradu slike, povećanje kašnjenja koje obrada slike unosi u sistem za prenos digitalne slike, kao i povećanje zahtevanog propusnog opsega kanala za prenos digitalnih slika. Stoga su unapređenje postojećih i razvoj novih metoda za kompresiju i dekompresiju digitalne slike jedini odgovor na izazove koji su postavljeni povećanjem zahteva za kvalitetom digitalne slike. Unapređivanjem postojećih i razvojem novih metoda potrebno je obezbediti, kako za kompresiju i dekompresiju digitalne slike sa gubicima, tako i za kompresiju i dekompresiju digitalne slike bez gubitaka, povećanje stepena kompresije, smanjenje utroška logičkih i memorijskih resursa i smanjenje kašnjenja.



Slika 1.1. Komunikacioni sistem koji upotrebljava kompresiju i dekompresiju slike prema postojećem stanju tehnike

Blok dijagram komunikacionog sistema koji upotrebljava kompresiju i dekompresiju digitalne slike prema postojećem stanju tehnike je prikazan na slici 1.1. Jednu od ključnih karika u sistemu za kompresiju i dekompresiju slike predstavlja dekodir digitalne slike, čija je uloga da izvrši dekompresiju digitalne slike (sa ili bez gubitaka). Blok dijagram dekodera za dekompresiju slike prema postojećem stanju tehnike je prikazan na slici 1.2. Unutar dekodera digitalne slike ulazna komprimovana

slika se prima od strane entropijskog dekodera koji svoje izlazne podatke prosleđuje dekoderskom procenjivaču verovatnoće. Dekoderski procenjivač verovatnoće vrši rekonstrukciju verovatnoća simbola unutar specificiranog konteksta, koje potom vraća natrag u entropijski dekodera. Istovremeno, izlazni podaci dekoderskog procenjivača verovatnoće se prosleđuju i dekvantizatoru, koji na osnovu njih generiše dekvantizovane odbirke signala (u slučaju kompresije digitalne slike sa gubicima) ili ih samo prosleđuje inverznom podopsežnom transformatoru (u slučaju kompresije bez gubitaka). Inverzni podopsežni transformator vrši kompoziciju i inverzno filtriranje odbiraka signala, generišući na svom izlazu piksele izlazne dekomprimovane slike. Svaka od navedenih komponenti dekodera utiče na stepen kompleksnosti, stepen utroška logičkih i memorijskih resursa i nivo kašnjenja koje u sistem za kompresiju i dekompresiju slike unosi dekodera.



Slika 1.2. Blok dijagram dekodera za dekompresiju digitalne slike prema postojećem stanju tehnike

U ovoj disertaciji je opisana hardverska realizacija novog brzog hardverskog dekodera komprimovane slike sa optimizovanim utroškom logičkih i memorijskih resursa, kao i minimizacijom kašnjenja koje produkuje sistem za kompresiju i dekompresiju digitalne slike, koga čine već razvijeni hardverski enkoder i brzi hardverski dekodera opisan u ovoj disertaciji. Realizacija brzog hardverskog dekodera je verifikovana implementacijom unutar jednog programabilnog logičkog kola nižeg cenovnog razreda, bez korišćenja bilo kakve spoljašnje memorije. Svaki od blokova koji sačinjavaju dekodera za dekompresiju slike (entropijski dekodera, dekoderski procenjivač verovatnoće, dekvantizator, inverzni podopsežni transformator) je optimizovan sa ciljem da ceo dekodera bude implementiran u programabilnom logičkom kolu uz minimizovan utrošak logičkih i memorijskih resursa.

Entropijski dekodler je realizovan predstavljanjem odgovarajućih parametara entropijskog dekodera preko stepena broja dva, čime je omogućena implementacija delitelja preko bitskih pomerača udesno. Sa druge strane umesto množača korišćene su *look-up* tabele. Ovim postupcima je značajno uprošćena hardverska realizacija entropijskog dekodera i postignuta ušteda u logičkim resursima u odnosu na realizacije entropijskog dekodera prema postojećem stanju tehnike.

S obzirom da postojeće dvoprolazne i višeprolazne metode zahtevaju privremeno smeštanje velike količine podataka u memoriju za potrebe procenjivanja verovatnoće, dekoderski procenjivač verovatnoće je zasnovan na adaptivnim histogramima koji omogućavaju postizanje velikog stepena kompresije korišćenjem jednoprolaznog algoritma, što je omogućilo značajnu uštedu memorijskih resursa. Korišćeni su koeficijenti koji se mogu predstaviti stepenom broja dva, čime je omogućeno realizovanje delitelja pomoću bitskih pomerača udesno što je rezultovalo značajnim uštedama u upotrebljenim logičkim resursima. Optimizacija utrošenih logičkih i memorijskih resursa pri izračunavanju konteksta za odbirke signala koji će biti raspoloživi na ulazu inverznog podopsežnog transformatora je postignuta na tako što je kontekst posmatranog odbirka signala izračunavan na osnovu susednih odbiraka signala uz posebno tretiranje magnitude i znaka za nenulte koeficijente i posebno tretiranje magnituda za nulte koeficijente kojih u praktičnoj realizaciji ima najviše.

Dekvantizacija odbiraka signala se vrši samo u slučaju dekompresije slike sa gubicima, dok se u slučaju dekompresije slike bez gubitaka odbirci signala direktno propuštaju kroz dekvantizator. U ovoj disertaciji je, kao najoptimalnije rešenje, izabran i implementiran uniformni skalarni dekvantizator sa mrtvom zonom.

Inverzni podopsežni transformator je realizovan na način da vrši dvodimenzionalnu diskretnu *wavelet* transformaciju korišćenjem *Le Gall*-ovih inverznih 5/3 filtara, koji omogućavanju savršenu rekonstrukciju signala. U disertaciji je predložena potpuno nova hardverska realizacija za dvodimenzionalnu inverznu diskretnu *wavelet* transformaciju sa *Le Gall*-ovim 5/3 filtrima, koja koristi manju količinu memorijskih resursa od bilo koje do sada predložene realizacije. Predložena realizacija omogućava da inverzno filtriranje na sledećem nivou kompozicije započne čim minimalan potreban broj odbiraka signala sa prethodnog nivoa kompozicije bude izgenerisan (tj. bez čekanja da kompletno filtriranje prethodnog nivoa bude završeno),

čime se postiže da se u toku filtriranja privremeno u memoriji pamte svega po dve linije odbiraka signala i odgovarajućih međurezultata za svaki od nivoa kompozicije. Uštedi memorijskih resursa doprinosi i to što inverzno horizontalno filtriranje započinje čim minimalan potreban broj odbiraka signala bude izgenerisan inverznim vertikalnim filtriranjem. Predložena je i nova, vremenski nestacionarna, hardverska realizacija inverznih horizontalnih filtara, koja značajno doprinosi uštedi logičkih resursa.

U izradi ove doktorske disertacije su primenjene kombinovane teorijske, simulacione i eksperimentalne metode u cilju identifikovanja i rešavanja problema u postojećim realizacijama dekodera komprimovane slike prema postojećem stanju tehnike. Izvršena je analiza problema i nedostataka pojedinih delova postojećih dekodera na bazi koje su određeni načini za prevazilaženje tih nedostataka. Simulacijama je ispitana i potvrđena funkcionalna ispravnost novih rešenja, kako za pojedine delove novog brzog dekodera, tako i za dekodera u celini, dok je sintezom dizajna određena količina utrošenih logičkih i memorijskih resursa. Eksperimentalna verifikacija hardverske realizacije novog brzog dekodera je izvršena implementacijom kompletnog dekodera u programabilnom čipu nižeg cenovnog razreda, bez korišćenja bilo kakvih spoljašnjih memorija. Eksperimentalnim putem je određeno i kašnjenje koje proizvodi kaskadna veza hardverskog enkodera i brzog hardverskog dekodera pri obradi video sekvenci.

Doktorska disertacija sadrži ukupno sedam poglavlja, uključujući uvod kao prvo poglavlje i zaključak kao sedmo poglavlje. U drugom poglavlju su opisani entropijski dekodera i dekoderski procenjivač verovatnoće, koji su deo novog brzog dekodera komprimovane slike, kao i njihova hardverska realizacija. Treće poglavlje sadrži opis dekvantizatora i njegove hardverske realizacije. U četvrtom poglavlju je dat opis inverznog podopsežnog transformatora i opis njegove hardverske realizacije. U petom poglavlju je predstavljena hardverska realizacija brzog dekodera za jednu ravan boje, dok je u šestom poglavlju opisana hardverska realizacija i implementacija brzog dekodera sa minimalnim utrošenim resursima za sliku u boji.

2. Entropijski dekodler i dekodlerski procenjivač verovatnoće

2.1. Uvod

Prilikom postupka kompresije digitalne slike u cilju postizanja većih stepena kompresije, simboli se često definišu na osnovu modela konteksta, tj. na osnovu susednih odbiraka signala nastalih postupkom direktne podopsežne transformacije. Opisi nultih stabala (tzv. *zerotrees*) i ostalih kontekstnih modela sa *wavelet* koeficijentima su dati u [1-4].

Iz literature je poznato da je moguće pronaći bitski kod, koji je efikasniji od koda fiksne dužine, ukoliko su poznate verovatnoće pojavljivanja pojedinih simbola. Dodela kodnih reči se obično vrši kodovanjem promenljive dužine, zatim *run-length* kodovanjem, Hafmanovim kodovanjem i aritmetičkim kodovanjem. Tehnike za eliminaciju alfabetske redundantnosti se uglavnom oslanjaju na prefiksne kodove i uglavnom vrše transformaciju poruka u bitske nizove na način da simbolima sa manjom verovatnoćom dodeljuju duže kodove, a simbolima sa većom verovatnoćom dodeljuju kraće kodove, kao što je to opisano u [5-7].

Najveći stepen kompresije se postiže aritmetičkim kodovanjem, koje teorijski može da ukloni svu redundantnu informaciju iz digitalizovane poruke, kao što se to može videti u [8-10]. Aritmetički *Q-coder* je prezentovan u [11-14], dok je aritmetički *Z-coder* opisan u [15-18].

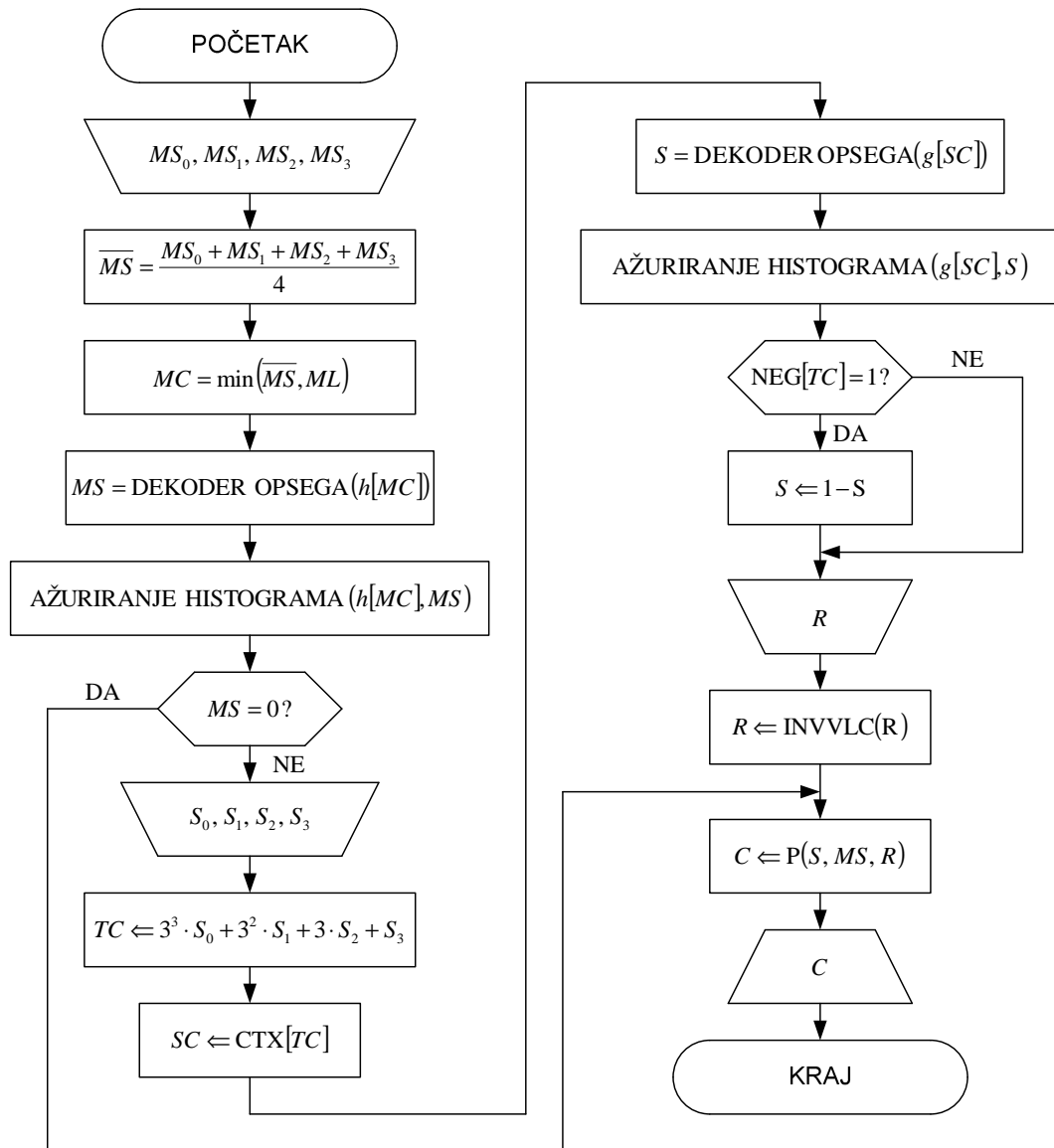
U poznatom standardu za kompresiju mirne slike JPEG 2000, koristi se *MQ* aritmetički koder koji je sličan *QM* koderu usvojenom u originalnom JPEG standardu opisanom u [19-21].

Kao osnova hardverske realizacije u ovoj disertaciji iskorišćen je postupak dekodovanja inverzan postupku kodovanja koji vrši koder opsega opisan u [22]. S obzirom da se enkodovanje odbiraka signala vrši posebno za svaki nivo dekompozicije i posebno za svaki podopseg, nezavisno od drugih nivoa dekompozicije i drugih podopsega, i proces dekodovanja se za svaki nivo kompozicije i svaki podopseg vrši nezavisno i odvojeno od ostalih nivoa kompozicije i ostalih podopsega.

2.2. Algoritam rada entropijskog dekodera i dekoderskog procenjivača verovatnoće

Dijagram toka entropijskog dekodera i dekoderskog procenjivača verovatnoće, zasnovan na jednoprolaznim adaptivnim histogramima, je prikazan na slici 2.1. Adaptacija započinje od uniformne raspodele i zahteva nekoliko odbiraka da bi se završila. Vreme adaptacije je proporcionalno broju odeljaka histograma i razlici između uniformne raspodele i stvarne raspodele promenljive koja se dekoduje.

Najpre se učitavaju vrednosti susednih indeksa amplitudne grupe MS_i (prikazanih na slici 2.2) već dekodovanih odbiraka komponenti dekomponovanog signala, na osnovu kojih se potom izračunava srednja vrednost susednih indeksa amplitudne grupe \overline{MS} . Potom se na osnovu izračunate vrednosti \overline{MS} određuje kontekst amplitude MC , koji predstavlja indeks odgovarajućeg adaptivnog histograma amplitude $h[MC]$ koji se zatim koristi za dekodovanje indeksa amplitudne grupe MS pomoću dekodera opsega. Kontekst amplitude MC je ograničen konstantom ML , sa preferencijalnom vrednošću $ML = 4$, iz razloga što lokalna varijansa može značajno da poraste blizu oštrih ivica na slici što bi dovelo do velikog broja histograma i njihove spore adaptacije. Broj histograma amplitude MH , tj. broj različitih konteksta amplitude MC , je preferencijalno ograničen na $MH = ML + 1 = 5$. Nakon dekodovanja indeksa amplitudne grupe MS vrši se ažuriranje histograma amplitude $h[MC]$.



Slika 2.1. Dijagram toka entropijskog dekodera i dekoderskog procenjivača verovatnoće

MS_0	MS_1	MS_2
MS_3	MS	

Slika 2.2. Susedni indeksi amplitudne grupe MS_i već dekodovanih odbiraka komponenti dekomponovanog signala

S_0	S_1	S_2
S_3	S	

Slika 2.3. Susedni znakovi S_i već dekodovanih odbiraka komponenti dekomponovanog signala

Tabela 2.1. CTX tabela za prevođenje vrednosti ternarnog konteksta TC u kontekst znaka SC

S_0	S_1	S_2	S_3	TC	SC	S_0	S_1	S_2	S_3	TC	SC	S_0	S_1	S_2	S_3	TC	SC
0	0	0	0	0	0	1	0	0	0	27	2	2	0	0	0	54	1
0	0	0	1	1	0	1	0	0	1	28	1	2	0	0	1	55	0
0	0	0	2	2	0	1	0	0	2	29	2	2	0	0	2	56	0
0	0	1	0	3	4	1	0	1	0	30	0	2	0	1	0	57	1
0	0	1	1	4	1	1	0	1	1	31	1	2	0	1	1	58	0
0	0	1	2	5	2	1	0	1	2	32	1	2	0	1	2	59	2
0	0	2	0	6	1	1	0	2	0	33	1	2	0	2	0	60	0
0	0	2	1	7	0	1	0	2	1	34	0	2	0	2	1	61	0
0	0	2	2	8	1	1	0	2	2	35	0	2	0	2	2	62	0
0	1	0	0	9	0	1	0	0	0	36	1	2	0	0	0	63	0
0	1	0	1	10	1	1	1	0	1	37	1	2	1	0	1	64	0
0	1	0	2	11	1	1	1	0	2	38	1	2	1	0	2	65	0
0	1	1	0	12	1	1	1	1	0	39	0	2	1	1	0	66	0
0	1	1	1	13	0	1	1	1	1	40	1	2	1	1	1	67	1
0	1	1	2	14	0	1	1	1	2	41	0	2	1	1	2	68	1
0	1	2	0	15	0	1	1	2	0	42	1	2	1	2	0	69	0
0	1	2	1	16	4	1	1	2	1	43	0	2	1	2	1	70	0
0	1	2	2	17	0	1	1	2	2	44	0	2	1	2	2	71	3
0	2	0	0	18	4	1	2	0	0	45	3	2	2	0	0	72	0
0	2	0	1	19	2	1	2	0	1	46	0	2	2	0	1	73	4
0	2	0	2	20	2	1	2	0	2	47	1	2	2	0	2	74	3
0	2	1	0	21	1	1	2	1	0	48	1	2	2	1	0	75	2
0	2	1	1	22	0	1	2	1	1	49	2	2	2	1	1	76	2
0	2	1	2	23	3	1	2	1	2	50	2	2	2	1	2	77	0
0	2	2	0	24	0	1	2	2	0	51	0	2	2	2	0	78	2
0	2	2	1	25	0	1	2	2	1	52	0	2	2	2	1	79	1
0	2	2	2	26	1	1	2	2	2	53	1	2	2	2	2	80	0

U slučaju kada je $MS = 0$, znak S se uopšte ne dekoduje. U slučaju kada je $MS \neq 0$, susedne vrednosti znakova S_i (prikazanih na slici 2.3) već dekodovanih odbiraka komponenti dekomponovanog signala se najpre učitavaju, a zatim se na osnovu njih dekoduje ternarni kontekst TC .

Tabela 2.2. NEG tabela za inverziju znaka S

TC	$P(0)$	$P(1)$	NS	TC	$P(0)$	$P(1)$	NS	TC	$P(0)$	$P(1)$	NS
0	0.5276	0.4724	0	27	0.6147	0.3853	0	54	0.4168	0.5832	1
1	0.5333	0.4667	0	28	0.4170	0.5830	1	55	0.5012	0.4988	0
2	0.4901	0.5099	1	29	0.6326	0.3674	0	56	0.5302	0.4698	0
3	0.2961	0.7039	1	30	0.4889	0.5111	1	57	0.5467	0.4533	0
4	0.4321	0.5679	1	31	0.4176	0.5824	1	58	0.5061	0.4939	0
5	0.6300	0.3700	0	32	0.4469	0.5531	1	59	0.4039	0.5961	1
6	0.4463	0.5537	1	33	0.5505	0.4495	0	60	0.5024	0.4976	0
7	0.4754	0.5246	1	34	0.5240	0.4760	0	61	0.4613	0.5387	1
8	0.4397	0.5603	1	35	0.4731	0.5269	1	62	0.4837	0.5163	1
9	0.5012	0.4988	0	36	0.4299	0.5701	1	63	0.5106	0.4894	0
10	0.5796	0.4204	0	37	0.5880	0.4120	0	64	0.5440	0.4560	0
11	0.4117	0.5883	1	38	0.5806	0.4194	0	65	0.5343	0.4657	0
12	0.5842	0.4158	0	39	0.4698	0.5302	1	66	0.4918	0.5082	1
13	0.5457	0.4543	0	40	0.4119	0.5881	1	67	0.4521	0.5479	1
14	0.5364	0.4636	0	41	0.5193	0.4807	0	68	0.5841	0.4159	0
15	0.5243	0.4757	0	42	0.4539	0.5461	1	69	0.5211	0.4789	0
16	0.7224	0.2776	0	43	0.4871	0.5129	1	70	0.4783	0.5217	1
17	0.5050	0.4950	0	44	0.4953	0.5047	1	71	0.6651	0.3349	0
18	0.7235	0.2765	0	45	0.3502	0.6498	1	72	0.4561	0.5439	1
19	0.3963	0.6037	1	46	0.4688	0.5312	1	73	0.6998	0.3002	0
20	0.6019	0.3981	0	47	0.5802	0.4198	0	74	0.6531	0.3469	0
21	0.4508	0.5492	1	48	0.4432	0.5568	1	75	0.6163	0.3837	0
22	0.5286	0.4714	0	49	0.3927	0.6073	1	76	0.5956	0.4044	0
23	0.6598	0.3402	0	50	0.6199	0.3801	0	77	0.5022	0.4978	0
24	0.4770	0.5230	1	51	0.5357	0.4643	0	78	0.6148	0.3852	0
25	0.5417	0.4583	0	52	0.4830	0.5170	1	79	0.4368	0.5632	1
26	0.4398	0.5602	1	53	0.4464	0.5536	1	80	0.5065	0.4935	0

Na osnovu ternarnog konteksta TC se potom određuje kontekst znaka SC pomoću CTX tabele prikazane u vidu tabele 2.1. CTX tabela prevodi 81 različitu vrednost ternarnog konteksta TC u preferencijalni broj od 5 različitih vrednosti konteksta znaka SC za svaki od podopsega (jer bi veliki broj različitih vrednosti konteksta znaka SC doveo do histograma koji se uopšte ne adaptiraju), što istovremeno predstavlja i broj histograma znaka SH . Ovaj veoma mali broj je opravdan iz razloga što se vrši dekodovanje verovatnijeg znaka S što je obezbeđeno odgovarajućim ispitivanjem znaka i , ako je potrebno, inverzijom znaka S pomoću tabele NEG prikazane u vidu tabele 2.2. Ternarni konteksti TC sa $NS = \text{NEG}[TC] = 0$ su odgovarajući većoj verovatnoći pozitivnog znaka $P(0)$ od verovatnoće negativnog znaka $P(1)$. Ternarni konteksti TC sa $NS = \text{NEG}[TC] = 1$ su odgovarajući većoj verovatnoći negativnog znaka $P(1)$ od verovatnoće pozitivnog znaka $P(0)$.

Kontekst znaka SC predstavlja indeks odgovarajućeg adaptivnog histograma znaka $g[SC]$ koji se zatim koristi za dekodovanje znaka S pomoću dekodera opsega. Nakon dekodovanja znaka S vrši se ažuriranje histograma znaka $g[SC]$.

Potom sledi učitavanje enkodovane vrednosti ostatka R , na bazi koje se pomoću dekodera sa promenljivom dužinom koda INV VLC vrši dekodovanje ostatka R . Na osnovu već dekodovanih vrednosti indeksa amplitudne grupe MS se, korišćenjem tabele 2.3 date za 16-bitne vrednosti odbiraka komponenti dekomponovanog signala, određuju donje granice odbiraka komponenti dekomponovanog signala koje se zatim u skladu sa jednačinom (2.1) sabiraju sa dekodovanim ostatkom R u cilju formiranja dekodovane vrednosti amplitude R . Konačno, na samom kraju postupka dekodovanja, dekodovana vrednost odbirka komponente dekomponovanog signala C se formira na osnovu već dekodovanih vrednosti amplitude M i znaka S .

$$M = R + \text{donja granica odbirka komponente} \quad (2.1)$$

Tabela 2.3. Dekodovanje granice odbiraka komponente dekomponovanog signala

Granice odbiraka komponenti (uključne)		Opseg odbiraka komponenti	<i>MS</i>
Niža	Viša		
0	0	1	0
1	1	1	1
2	2	1	2
3	3	1	3
4	5	2	4
6	7	2	5
8	11	4	6
12	15	4	7
16	23	8	8
24	31	8	9
32	47	16	10
48	63	16	11
64	95	32	12
96	127	32	13
128	191	64	14
192	255	64	15
256	383	128	16
384	511	128	17
512	767	256	18
768	1023	256	19
1024	1535	512	20
1536	2047	512	21
2048	3071	1024	22
3072	4095	1024	23
4096	6143	2048	24
6144	8191	2048	25
8192	12287	4096	26
12288	16383	4096	27
16384	24575	8192	28
24576	32767	8192	29
32768	49151	16384	30
49152	65535	16384	31

Pritom treba naglasiti da je prilikom kodovanja komponente dekomponovanog signala C u postupku kompresije slike, znak S formiran na način:

$$S = \begin{cases} 0, & C > 0 \\ 2, & C = 0, \\ 1, & C < 0 \end{cases} \quad (2.2)$$

amplituda M određena kao (slično postupku objašnjenom u [23]):

$$M = |C|, \quad (2.3)$$

a ostatak R kao:

$$R = M - \text{donja granica odbiraka komponente}, \quad (2.4)$$

pri čemu su granice odbiraka komponenti dekomponovanog signala i indeks amplitudne grupe MS , određeni na osnovu tabele 2.3.

Tabela 2.4. Ukupni kapacitet zahtevane memorije za smeštanje informacija o kontekstu

Nivo kompozicije	Kapacitet zahtevane memorije
0	$\frac{W}{2}$
1	$\frac{W}{4}$
2	$\frac{W}{8}$
n	$\frac{W}{2^{n+1}}$
$N-1$	$\frac{W}{2^N}$
Svi nivoi	$\sum_{n=0}^{N-1} \frac{W}{2^n} = W \cdot \left(1 - \frac{1}{2^N}\right) < W$
Svi nivoi i svi podopsezi	$3 \cdot \sum_{n=0}^{N-1} \frac{W}{2^n} = 3 \cdot W \cdot \left(1 - \frac{1}{2^N}\right) < 3 \cdot W$

Kapacitet zahtevane memorije za smeštanje informacija o kontekstu za sve podopsege (LH, HL i HH) u toku operacije dekodovanja je približno tri ekvivalentne linije slike, kao što se može videti iz tabele 2.4. Pritom, podopseg LL na najdubljem nivou kompozicije može biti direktno smešten u memoriju bez prethodnog enkodovanja u postupku kompresije, ili prethodno biti enkodovan korišćenjem bilo koje metode kompresije za male slike prema postojećem stanju tehnike. U zavisnosti od toga, koristi se i odgovarajuća procedura za dekodovanje.

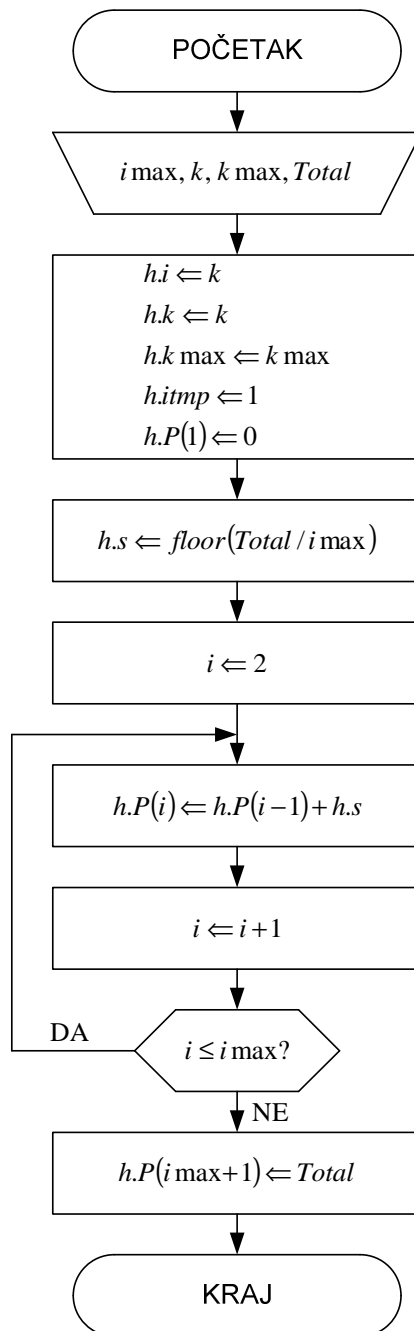
Dijagram toka inicijalizacije histograma sa brzom adaptacijom je prikazan na slici 2.4. Svaki od odeljaka histograma odgovara jednom simbolu x , koji može biti MS za histogram amplitude ili S za histogram znaka. Prema postojećem stanju tehnike, postupak procene verovatnoće $p(x)$ pojavljivanja simbola x zasnovan je na broju pojavljivanja $u(x)$ simbola x i broju pojavljivanja svih simbola $Total$. Takođe, moguće je definisati i kumulativnu verovatnoću $P(x)$ svih simbola y koji u alfabetu prethode simbolu x .

$$p(x) = \frac{u(x)}{Total}, \quad (2.5)$$

$$Total = \sum_x u(x), \quad (2.6)$$

$$P(x) = \sum_{y < x} p(y) = \frac{U(x)}{Total}, \quad (2.7)$$

$$U(x) = \sum_{y < x} u(y). \quad (2.8)$$



Slika 2.4. Dijagram toka inicijalizacije histograma sa brzom adaptacijom

Ovaj jednostavni postupak ima jedan značajan nedostatak. Naime, broj pojavljivanja svih simbola $Total$ je proizvoljan ceo broj, zbog čega neophodno primeniti operaciju deljenja prilikom određivanja verovatnoće $p(x)$. Iz tog razloga je u realizaciji koja je predmet ove disertacije usvojeno da bude:

$$Total = 2^{w_3}, \quad (2.9)$$

čime je omogućeno da operacija deljenja bude zamenjena operacijom pomeranja udesno za w_3 bita.

Drugi nedostatak opisanog postupka prema postojećem stanju tehnike je spora adaptacija verovatnoće $p(x)$ zbog procesa usrednjavanja. Zbog toga je kao rešenje u realizaciji koja je predmet ove disertacije usvojeno da se adaptacija verovatnoće $p(x)$ postiže niskopropusnim filtriranjem binarne sekvence $I(j)$ koja predstavlja pojavljivanje simbola x u sekvenci y .

$$I(j) = \begin{cases} 1, & y(j) = x \\ 0, & y(j) \neq x \end{cases} \quad (2.10)$$

Značajnu ulogu u ovom procesu ima vremenski odziv filtra propusnika niskih učestanosti, jer veća vremenska konstanta omogućava preciznije predviđanje u ustaljenom stanju, dok manja vremenska konstanta omogućava brže predviđanje. Ovaj problem posebno dolazi do izražaja na početku procesa adaptacije, kada je prisutan nedostatak informacija. Da se ne bi pravio kompromis pri izboru fiksne vrednosti dominantnog pola filtra, primenjeno je rešenje koje omogućava promenljivost dominantnog pola između minimalne i maksimalne vrednosti.

Prema dijagramu toka inicijalizacije histograma prikazanom na slici 2.4, najpre se učitavaju odgovarajuće vrednosti promenljivih, na osnovu kojih se potom inicijalizuju promenljive unutar strukture histograma h .

Na prikazanom dijagramu, parametar i predstavlja indeks odeljka histograma, koji može imati vrednosti u opsegu od 1 do i_{max} . Parametar i_{max} predstavlja maksimalnu vrednost indeksa i nepraznog odeljka histograma, tj. ukupan broj različitih simbola u alfabetu, koji je preferencijalno manji ili jednak 32 za histogram amplitude ili jednak 2 za histogram znaka. Parametar $h.P()$ predstavlja niz kumulativnih verovatnoća $h.P(i) = P(y | y < i) = \sum_{y < i} p(y)$. Parametar $h.k$ je recipročna vrednost apsolutnog dominantnog pola filtra propusnika niskih učestanosti. Menjanje njegove vrednosti

između $h.k \max$ i $h.k \min$ omogućava brzu adaptaciju histograma nakon početka rada. Parametar $h.k \max$ predstavlja recipročnu vrednost minimalnog apsolutnog dominantnog pola filtra propusnika niskih učestanosti, i reč je o fiksnom empirijskom parametru koji je manji od vrednosti parametra $Total$. Parametar $h.k \min$ predstavlja recipročnu vrednost maksimalnog apsolutnog dominantnog pola filtra propusnika niskih učestanosti, i reč je o fiksnom empirijskom parametru sa preferencijalnom vrednošću $h.k \min = 2$. Ukupan broj simbola unutar histograma uvećan za 1 je predstavljen parametrom $h.i$. Konačno, parametar $h.itmp$ predstavlja privremenu vrednost parametra $h.i$ pre promene parametra $h.k$.

Nakon inicijalizovanja promenljivih unutar strukture histograma h , u skladu sa dijagramom sa slike 2.4, sledi izračunavanje veličine koraka $h.s$, inicijalizovanje indeksa i i inicijalizovanje histograma. Potom sledi inkrementiranje indeksa i i ispitivanje njegove vrednosti. Poslednji korak predstavlja inicijalizaciju poslednjeg odeljka histograma.

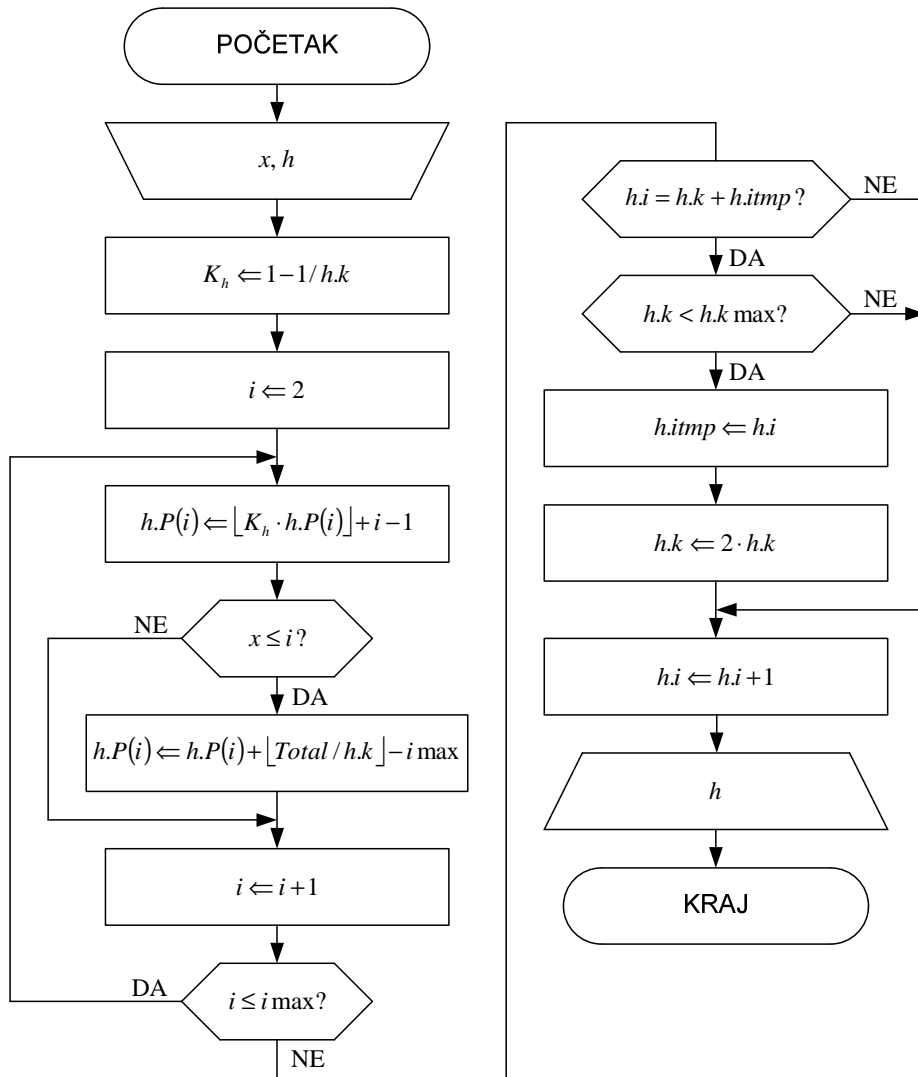
Na slici 2.5 je prikazan dijagram toka algoritma za ažuriranje histograma sa brzom adaptacijom, koji je zasnovan na ulazu simbola x i već objašnjennoj strukturi histograma h . S obzirom da dekođer opsega ne može da radi sa predviđenom nultom verovatnoćom $p(x)=0$, čak ni za simbole koji se uopšte ne pojavljuju, postoji potreba za modifikovanjem binarne sekvence $I(j)$. Još jedan razlog za modifikovanje binarne sekvence $I(j)$ leži u činjenici da se modifikovana verovatnoća $Mp(x) = Total \cdot p(x)$ predviđa primenom aritmetike u formatu sa fiksnom tačkom. Adaptacija verovatnoće $p(x)$ se vrši niskopropusnim filtriranjem modifikovane binarne sekvence $MI(j)$:

$$MI(j) = \begin{cases} Total - i \max, & y(j) = x \\ 1, & y(j) \neq x \end{cases} \quad (2.11)$$

Pri tome se maksimalna verovatnoća $\max p(x)$ i minimalna verovatnoća $\min p(x)$ mogu predstaviti kao:

$$\max p(x) = \frac{Total - i \max}{Total} < 1, \quad (2.12)$$

$$\min p(x) = \frac{1}{Total} > 0. \quad (2.13)$$



Slika 2.5. Dijagram toka algoritma za ažuriranje histograma sa brзом adaptacijom

Preferirani niskopropusni filter je IIR filter prvog reda kod koga se operacija deljenja eliminiše održavanjem parametra $h.k$ u vidu stepena broja 2 u toku promene:

$$Mp(x) \leftarrow Mp(x) \cdot \left(1 - \frac{1}{h.k}\right) + MI(j). \quad (2.14)$$

Umesto ažuriranja modifikovane verovatnoće $Mp(x)$, vrši se ažuriranje modifikovane kumulativne verovatnoće $MP(x) = Total \cdot P(x)$, tj. ažuriranje niza kumulativnih verovatnoća $h.P()$. Konstanta K_h , koja se koristi za brzu adaptaciju histograma, i indeks i odeljka histograma se najpre inicijalizuju. Potom se kumulativnoj verovatnoći $h.P(i)$ preskaliranoj sa konstantom K_h dodaje $i-1$, što je ekvivalentno dodavanju jedinice broju $u(x)$. Zatim sledi ažuriranje kumulativne verovatnoće $h.P(i)$ i to samo za histograme sa indeksom i većim ili jednakim od x što se utvrđuje prethodnim ispitivanjem vrednosti tih parametara.

U ostatku algoritma se vrši ažuriranje histograma koje je ekvivalentno sledećim matematičkim formulama:

$$h.k = \min \left[2^{\lfloor \log_2(h.i + h.k \min - 2) \rfloor}, h.k \max \right] \quad (2.15)$$

$$h.k = \max(h.k, h.k \min), \quad (2.16)$$

pri čemu je preferencijalna vrednost $h.k \min = 2$, što je od značaja za prvi $h.k$ u toku procesa brze adaptacije.

Opisani postupak brze adaptacije histograma ima značajne prednosti u poređenju sa postupcima prema postojećem stanju tehnike. Modifikacije procenjene verovatnoće su velike na početku procesa procene i mnogo manje posle toga, čime se omogućava detekcija malih lokalnih promena verovatnoće, što doprinosi povećanju stepena kompresije/dekompresije slike.

Zarad boljeg razumevanja algoritma dekodera opsega realizovanog u ovoj disertaciji, koji je kompatibilan sa enkoderom opsega opisanim u [24], najpre će ukratko biti opisani algoritam enkodera opsega prema postojećem stanju tehnike [25-27] i algoritam enkodera opsega iz [24].

Dijagram toka enkodera opsega prema postojećem stanju tehnike [25-27] je prikazan na slici 2.6.

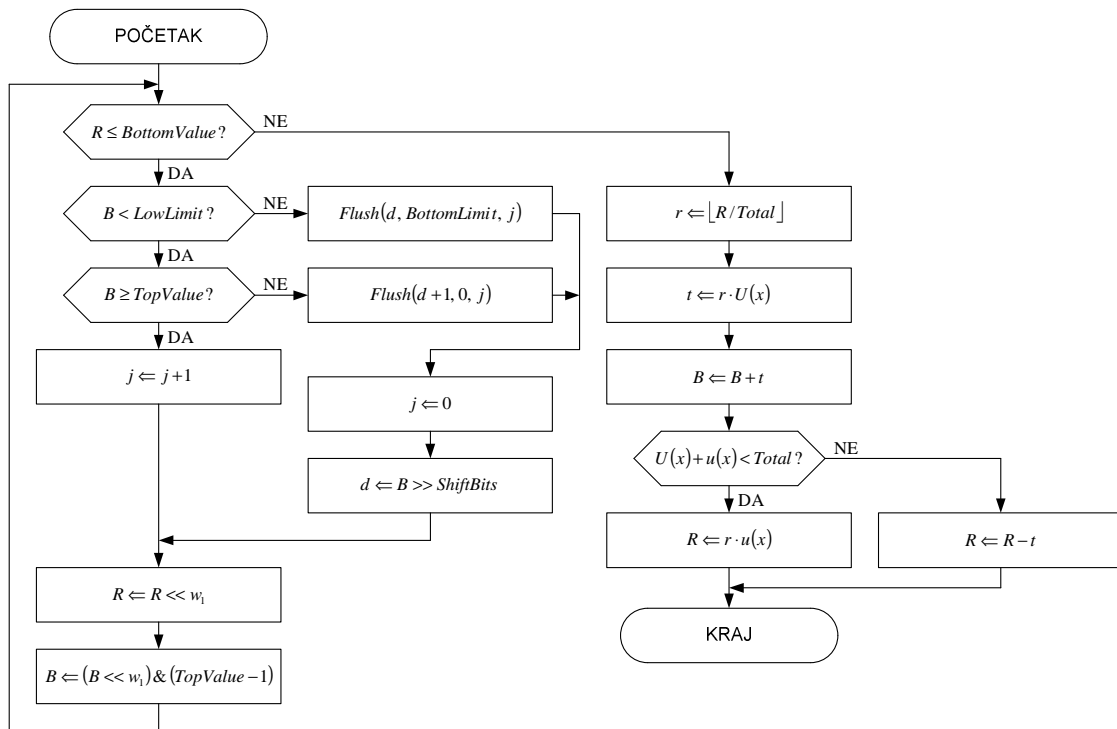
Simbol x se enkoduje u baferu čija je širina $s = b^w$ u vidu broja i :

$$i \in \left(\lfloor s \cdot P(x) \rfloor, \lfloor s \cdot (P(x) + p(x)) \rfloor \right), \quad (2.17)$$

$$\lfloor s \cdot P(x) \rfloor \leq i < \lfloor s \cdot (P(x) + p(x)) \rfloor, \quad (2.18)$$

$$s \cdot P(x) < i + 1 \leq s \cdot (P(x) + p(x)), \quad (2.19)$$

$$P(x) < \frac{i+1}{s} \leq P(x) + p(x). \quad (2.20)$$



Slika 2.6. Dijagram toka enkodera opsega prema postojećem stanju tehnike

Za predstavljanje stanja enkodera se koriste sledeće promenljive: T = gornja granica opsega, B = donja granica opsega, R = opseg (pri čemu je $R = T - B$), d = izlazni bajt i j = broj bajtova podkoračenja.

Algoritam enkodera opsega u formatu sa pokretnom tačkom posle renormalizacije i bez provere graničnih uslova je:

$$t \leftarrow R \cdot P(x), \quad (2.21)$$

$$B \Leftarrow B + t, \quad (2.22)$$

$$R \Leftarrow R \cdot p(x), \quad (2.23)$$

Nakon uvođenja preskaliranog opsega r , algoritam enkodera opsega u celobrojnoj aritmetici, posle renormalizacije i bez provere graničnih uslova je:

$$r \Leftarrow \left\lfloor \frac{R}{Total} \right\rfloor, \quad (2.24)$$

$$t \Leftarrow r \cdot U(x), \quad (2.25)$$

$$B \Leftarrow B + t, \quad (2.26)$$

$$R \Leftarrow r \cdot u(x). \quad (2.27)$$

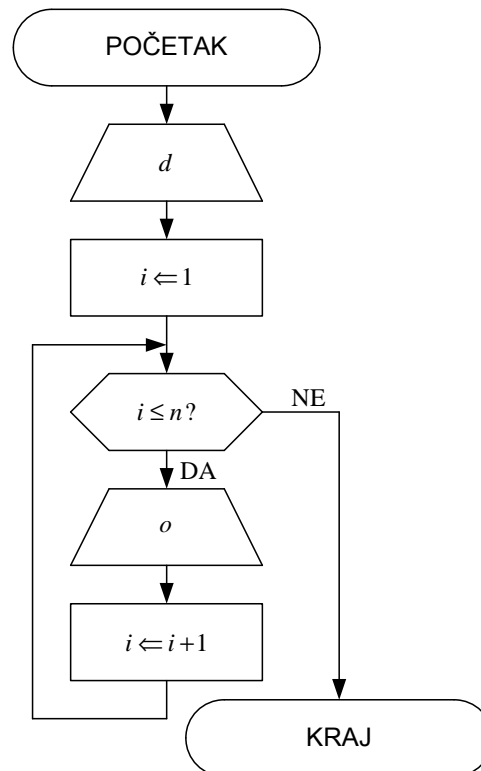
Na dijagramima pomoću kojih se opisuju algoritmi za enkoder i dekoder opsega, korišćeni su aritmetički operatori koji se inače koriste u višim programskim jezicima, kao što su npr. C ili C++, i to:

- $x \ll y$ = pomeraj x ulevo za y bita;
- $x \gg y$ = pomeraj x udesno za y bita;
- $x \% y$ = celobrojni ostatak deljenja x sa y ;
- $x | y$ = x ili y ;
- $x \& y$ = x i y .

Konstante koje figurišu na dijagramima pomoću kojih se opisuju algoritmi za enkoder i dekoder opsega: *TopValue*, *BottomValue*, *ShiftBits*, *ExtraBits*, *BottomLimit* i *LowLimit* imaju preferencijalne vrednosti koje zavise od dve konstante w_1 i w_2 čije su preferencijalne vrednosti 8 i 32, respektivno. Dakle, preferencijalne vrednosti pomenutih konstanti su:

- $TopValue = 1 \ll (w_2 - 1) = 40000000h$;

- $BottomValue = TopValue \gg w_1 = 00400000h$;
- $ShiftBits = w_2 - w_1 - 1 = 23$;
- $ExtraBits = (w_2 - 2) \% w_1 + 1 = 4$;
- $BottomLimit = (1 \ll w_1) - 1 = 0FFh$;
- $LowLimit \ll BottomLimit \ll ShiftBits$.



Slika 2.7. Dijagram toka za *flush* proceduru prema postojećem stanju tehnike

Konstanta *ExtraBits* se upotrebljava iz razloga što parametri B , T i s treba da imaju vrednosti predstavljene u formatu sa fiksnom tačkom, i to sa *ExtraBits* bita posle decimalne tačke. Značaj ovakve predstave pomenutih parametara se ogleda kod dekodera, pošto izbor *ExtraBits* u opsegu $(1, w_1)$ umesto često korišćenog opsega $(0, w_1 - 1)$ doprinosi smanjenju složenosti algoritma, jer bi u protivnom, bio potreban dodatni blok za ispitivanje nulte vrednosti *ExtraBits*. Pre početka procesa enkodovanja opsega, neophodno je da sledeće promenljive budu inicijalizovane i to na sledeći način:

- $B = 0$;

- $R = TopValue;$
- $d = 0;$
- $j = 0.$

Na slici 2.7 je prikazan dijagram toka za *flush* proceduru prema postojećem stanju tehnike, koja se koristi u sklopu dijagrama toka enkodera opsega prema postojećem stanju tehnike, koja najpre izbacuje bajt d , inicijalizuje brojač petlje i , testira i inkrementira brojač i unutar petlje i izbacuje j bajtova o .

U prvom delu algoritma za enkodovanje opsega koji je prikazan na slici 2.6, vrši se renormalizacija pre procesa enkodovanja. U slučaju opsega sa mogućim prenosom, prva *flush* procedura sa slike 2.7 izbacuje bajt d i sve bajtove podkoračenja $0FFh$. U slučaju opsega sa stvarnim prenosom, druga *flush* procedura izbacuje bajt $d+1$ i sve bajtove podkoračenja $0h$, jer $0FFh$ postaje $0h$ usled prenosa. U oba slučaja nakon *flush* procedure, vrši se inicijalizacija broja bajtova podkoračenja j i generisanje izlaznog bajta d , dodelom višeg bajta parametra B . Ažuriranje parametra R se vrši pomeranjem bajta koji je već bio izlaz ili koji će biti izlaz u slučaju $0FFh$. Ažuriranje parametra B se vrši pomeranjem i brisanjem bita prenosa.

U drugom delu algoritma za enkodovanje opsega koji je prikazan na slici 2.6, vrši se ažuriranje opsega. Najpre se vrši ažuriranje preskaliranog opsega r za sve simbole pomoću prve operacije deljenja. Zatim se izvodi opseg t tekućeg simbola korišćenjem prve operacije množenja sa $U(x)$ za tekući simbol x . Potom sledi ažuriranje parametra B , a zatim i ažuriranje parametra R korišćenjem druge operacije množenja sa $u(x)$ za tekući simbol x i to za sve simbole osim poslednjeg. Kada je reč o poslednjem simbolu, parametar R se ažurira korišćenjem operacije oduzimanja.

Ukupan broj iteracija u ovoj petlji u najgorem slučaju iznosi $\log_b(s)$, ali se smanjuje u slučaju izbora veće vrednosti b . To predstavlja najznačajniju prednost koju koder opsega prema postojećem stanju tehnike ima u odnosu na aritmetičke kodere prema postojećem stanju tehnike. Npr. za $s = 2^{31}$, aritmetički koder će koristiti $b = 2$, a dekoder opsega će koristiti $b = 2^8 = 256$, što znači da će u najgorem slučaju aritmetičkom koderu biti potrebne 32 iteracije po simbolu, dok će koder opsega zahtevati samo 4 iteracije po simbolu. Takođe, binarni aritmetički koderi prema postojećem stanju tehnike u većem broju slučajeva zahtevaju renormalizaciju opsega

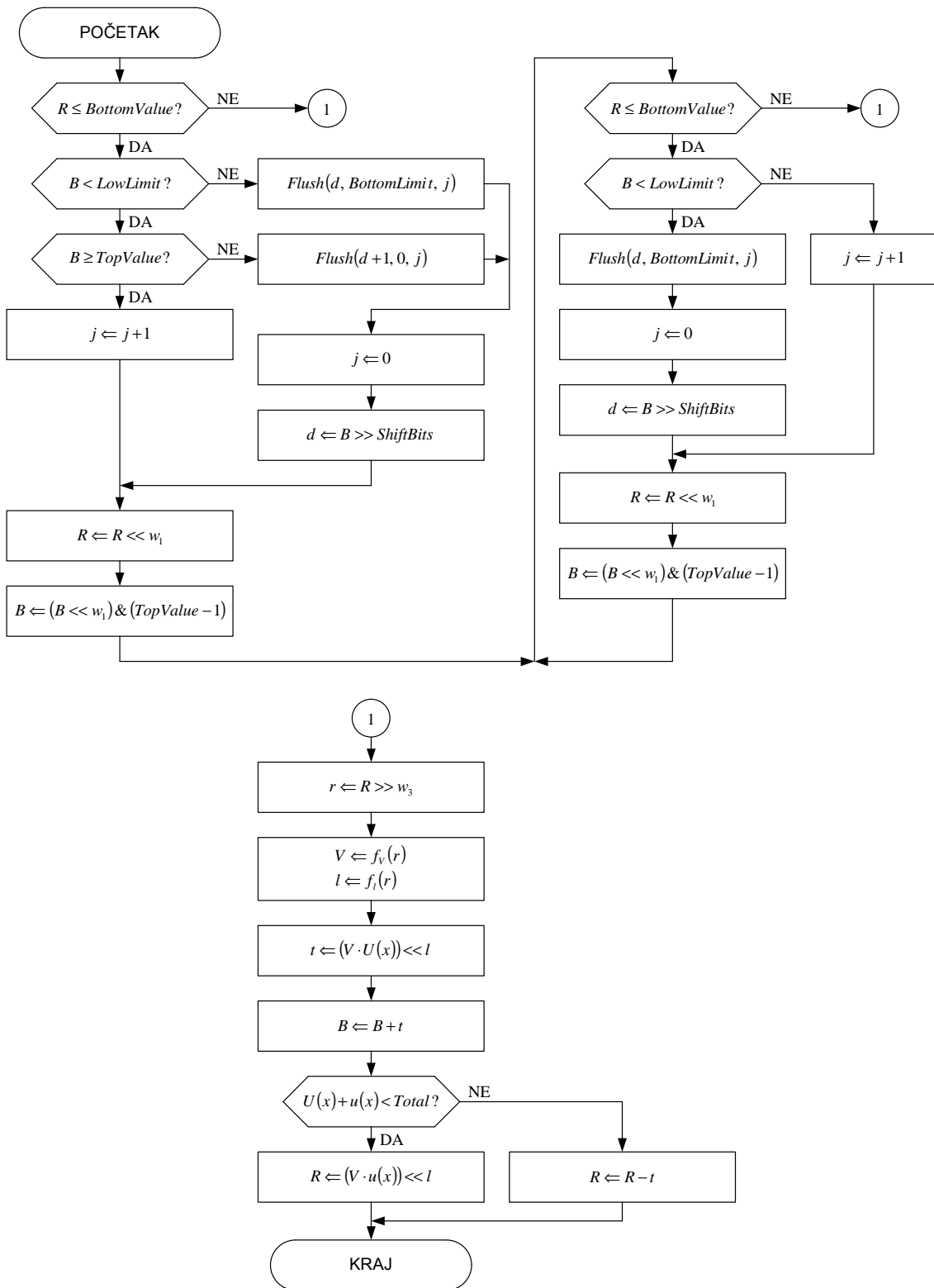
nego što je to slučaj sa koderom opsega prema postojećem stanju tehnike, što doprinosi skoro dvostruko manjoj brzini izvršavanja.

Na slici 2.8 je prikazan dijagram toka enkodera opsega, koji je opisan u [24] i sa kojim je kompatibilan dekoder opsega realizovan u ovoj disertaciji.

Pomenuti enkoder opsega ne sadrži operaciju deljenja i opciono, ne sadrži ni operaciju množenja. Prva operacija deljenja sa $Total = 2^{w_3}$ (prilikom izračunavanja vrednosti parametra r) se realizuje pomoću operacije pomeranja udesno za w_3 bita, zahvaljujući brznoj adaptaciji histograma. Parametar r se zatim predstavlja kao $r = V \cdot 2^l$, da bi se potom prva operacija množenja izvršila pomoću množenja sa malim brojem V i pomeranjem ulevo za l bita u cilju izračunavanja vrednosti parametra t . Druga operacija množenja se vrši prilikom izračunavanja vrednosti parametra R pomoću množenja sa malim brojem V i pomeranjem ulevo za l bita. Obe operacije množenja sa malim brojem V su značajno pojednostavljene zbog malog broja bita koji se koriste za predstavljanje broja V . Osim toga, množenje sa malim neparnim brojevima $V = 3$ ili $V = 5$ je moguće realizovati pomoću kombinacije operacija pomeranja i sabiranja, čime se u potpunosti eliminiše operacija množenja.

Na slici 2.9 je prikazan dijagram toka dekodera opsega prema postojećem stanju tehnike, koji je zajedno sa enkoderom opsega prema postojećem stanju tehnike opisan u [25-27]. Dekodovanje se obavlja pomoću tabele za pretraživanje LUT , koja je kompatibilna sa jednačinama (2.17)-(2.20) za enkodovanje simbola x :

$$x = LUT\left(\frac{i+1}{s}\right). \quad (2.28)$$



Slika 2.8. Dijagram toka enkodera opsega koji je realizovan u ovoj disertaciji

Algoritam dekodera opsega u formatu sa pokretnom tačkom posle renormalizacije i bez provere graničnih uslova je:

$$t \leftarrow B / R, \quad (2.29)$$

$$x \leftarrow LUT(t), \quad (2.30)$$

$$t \leftarrow R \cdot P(x), \quad (2.31)$$

$$B \leftarrow B - t, \quad (2.32)$$

$$R \leftarrow R \cdot p(x). \quad (2.33)$$

Nakon uvođenja preskaliranog opsega r , algoritam dekodera opsega u celobrojnoj aritmetici, posle renormalizacije i bez provere graničnih uslova je:

$$r \leftarrow \left\lfloor \frac{R}{Total} \right\rfloor, \quad (2.34)$$

$$t \leftarrow \left\lfloor \frac{B}{r} \right\rfloor, \quad (2.35)$$

$$x \leftarrow LUTr(t), \quad (2.36)$$

$$t \leftarrow r \cdot U(x), \quad (2.37)$$

$$B \leftarrow B - t, \quad (2.38)$$

$$R \leftarrow r \cdot u(x), \quad (2.39)$$

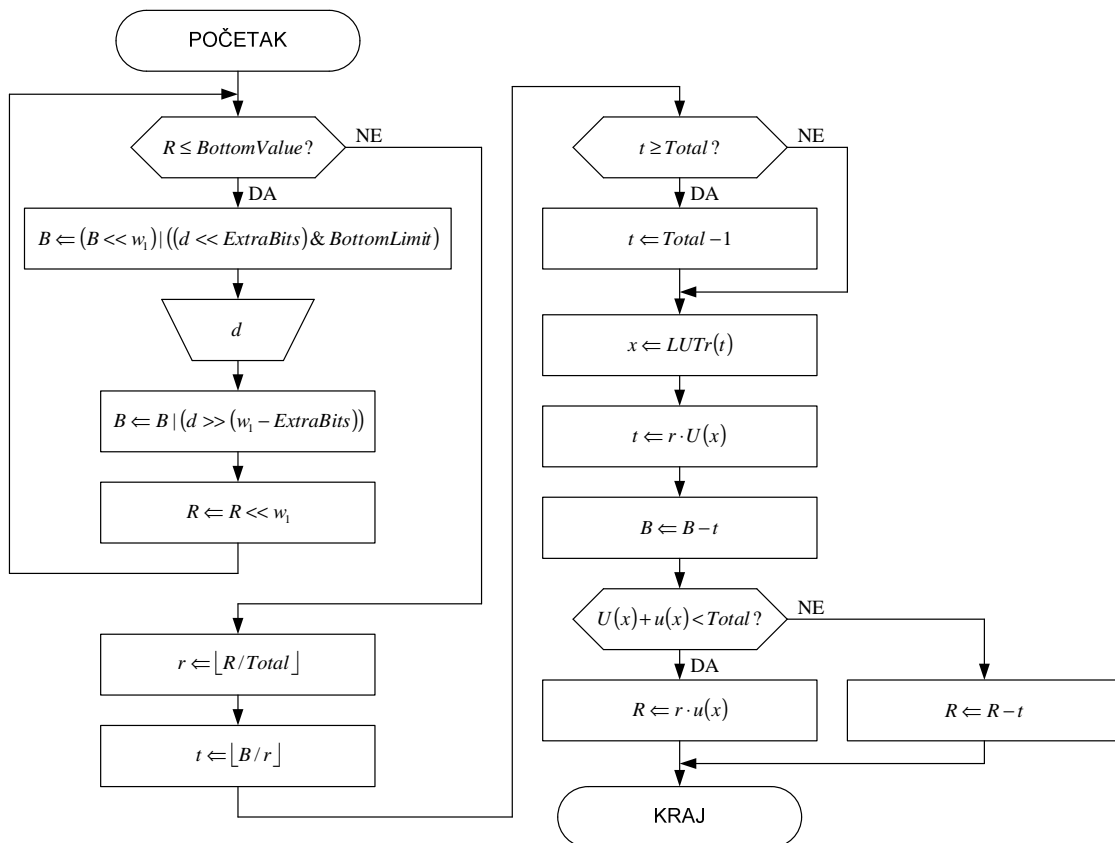
pri čemu je:

$$LUTr(t \cdot Total) = LUTr\left(\frac{B}{r}\right) = LUT(t). \quad (2.40)$$

Ulaz su cifre simbola x u osnovi b iz ulaznog bafera. Prvih $2w_1 - ExtraBits$ bita se ignorišu prema konceptu ekstra bita. U ovom konkretnom slučaju, prvi bajt je lažan. Pre početka procesa dekodovanja opsega, potrebno je inicijalizovati sledeće promenljive:

$$B = d \gg (w_1 - ExtraBits), \quad (2.41)$$

$$R = 1 \ll ExtraBits. \quad (2.42)$$

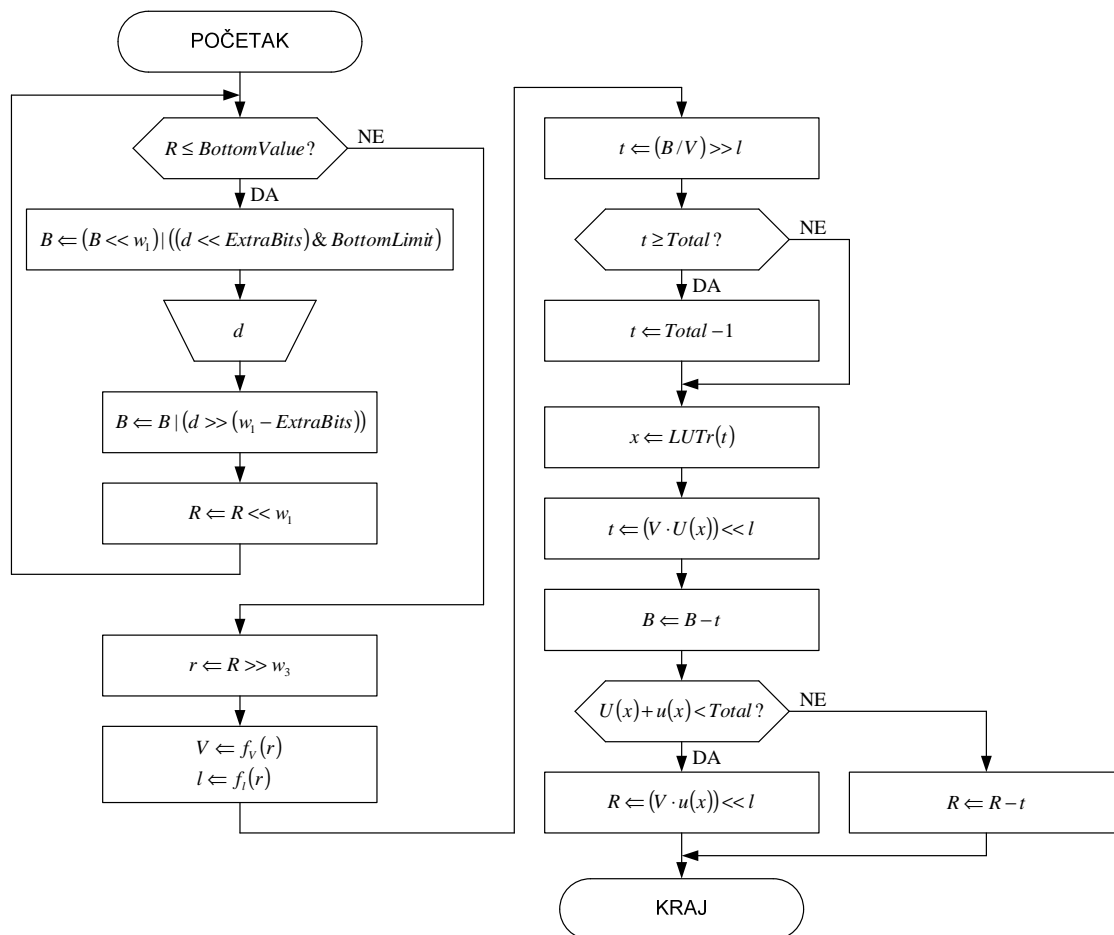


Slika 2.9. Dijagram toka dekodera opsega prema postojećem stanju tehnike

Prvi deo algoritma za dekodovanje opsega koji je prikazan na slici 2.9 vrši renormalizaciju pre dekodovanja u skladu sa početnim ispitivačkim blokom. Zatim sledi upisivanje odgovarajućih bita u parametar B i učitavanje novog simbola d . Nakon toga se ažurira parametar B pomoću odgovarajuće operacije pomeranja, da bi se potom izvršilo i ažuriranje parametra R pomoću operacije pomeranja.

Drugi deo algoritma za dekodovanje opsega koji je prikazan na slici 2.9 vrši ažuriranje opsega. Najpre se vrši ažuriranje preskaliranog opsega r za sve simbole korišćenjem prve operacije deljenja. Zatim sledi izvođenje kumulativnog broja pojavljivanja t tekućeg simbola pomoću druge operacije deljenja, a potom i ograničavanje vrednosti t u slučaju ispunjenja odgovarajućeg uslova. Sledeći korak je pronalaženje odgovarajućeg simbola x na osnovu vrednosti parametra t , a zatim i preskaliciranje vrednosti parametra t . Dalje se ažurira vrednost parametra B , a potom i vrednost parametra R pomoću druge operacije množenja sa $u(x)$ za tekući simbol x i to za sve simbole osim poslednjeg. U slučaju poslednjeg simbola, parametar R se ažurira pomoću operacije oduzimanja. Po završetku dekodovanja svih podataka, izvršava se i poslednja renormalizacija.

U dekoderu opsega prema postojećem stanju tehnike prvu operaciju deljenja sa $Total$ je moguće realizovati pomoću pomeranja udesno za w_3 bita u slučaju kada je $Total = 2^{w_3}$, što i jeste obezbeđeno pomoću dekoderskog procenjivača verovatnoće. Međutim druga operacija deljenja se ne može eliminisati, što doprinosi povećanju složenosti dekoderskog procesora jer veliki broj postojećih digitalnih procesora signala ne podržava operaciju deljenja. Osim toga u dekoderu opsega postoje i dve operacije množenja po svakom simbolu komprimovane slike, što doprinosi smanjivanju brzine obrade u mikroprocesorima opšte namene. Upravo ovi nedostaci su otklonjeni u dekoderu opsega opisanom u ovoj disertaciji.



Slika 2.10. Dijagram toka dekodera opsega koji je realizovan u ovoj disertaciji

Na slici 2.10 je prikazan dijagram toka dekodera opsega koji je realizovan u ovoj disertaciji, bez operacije deljenja i opciono, bez operacije množenja. Prva operacija deljenja sa $Total = 2^{w_3}$ (prilikom izračunavanja vrednosti parametra r) se realizuje pomoću operacije pomeranja udesno za w_3 bita, zahvaljujući brznoj adaptaciji histograma opisanoj u ovoj disertaciji. Parametar r se zatim predstavlja kao $r = V \cdot 2^l$, da bi se potom prva operacija množenja izvršila pomoću množenja sa malim brojem V i pomeranjem ulevo za l bita u cilju izračunavanja vrednosti parametra t . Druga operacija množenja se vrši prilikom izračunavanja vrednosti parametra R pomoću množenja sa malim brojem t i pomeranjem ulevo za l bita. Obe operacije množenja sa malim brojem V su značajno pojednostavljene zbog malog broja bita koji se koriste za predstavljanje broja V . Osim toga, množenje sa malim neparnim brojevima $V = 3$ ili

$V = 5$ je moguće realizovati pomoću kombinacije operacija pomeranja i sabiranja, čime se u potpunosti eliminiše operacija množenja. Druga operacija deljenja sa r (prilikom izračunavanja vrednosti parametra t) se realizuje pomoću deljenja sa malim brojem V i pomeranjem udesno za l bita. Pritom, operacija deljenja sa konstantnim malim neparnim brojevima 3, 5, 9, 11, 13 ili 15 se može realizovati pomoću jedne operacije množenja i jednog pomeranja udesno u skladu sa tabelom 2.5, kao što je to pokazano u [28,29]. Pritom, operacija deljenja sa 7 je najsloženija jer zahteva sprovođenje operacije sabiranja sa $049240249h$ i operacije sabiranja sa prenosom i $0h$ između operacija množenja i pomeranja udesno koje su prikazane u tabeli 2.5.

Tabela 2.5. Realizacija operacije deljenja sa brojevima 3, 5, 7, 9, 11, 13, i 15

Deljenje sa [decimalni broj]	Množenje sa [heksadecimalni broj]	Pomeraj udesno za [binarnih cifara]
3	0AAAAAAAAAB	1
5	0CCCCCCCD	2
7	049249249	1
9	038E38E39	1
11	0BA2E8BA3	3
13	04EC4EC4F	2
15	088888889	3

Aproksimacije koje se koriste u realizaciji operacija množenja ili deljenja u ovoj disertaciji uzrokuju izvesno smanjivanje stepena kompresije/dekompresije. Npr. izborom $V = 1$ je moguće u potpunosti ukloniti sve operacije množenja i deljenja, ali se time postiže i najveća greška aproksimacije i najveće smanjenje stepena kompresije/dekompresije (ne više od 5%). Ako se, pak, dozvoli da V uzima vrednosti 1 ili 3, stepen kompresije se veoma malo smanjuje (manje od 1%). U tabelama 2.6 i 2.7 je prikazana razlika u broju operacija množenja i deljenja po dekodovanom simbolu između dekodera opsega prema postojećem stanju tehnike (STARI DEKODER) i dekodera opsega koji je opisan u ovoj disertaciji (NOVI DEKODER).

Tabela 2.6. Broj operacija množenja i deljenja po dekodovanom simbolu za $Total \neq 2^{w_3}$

$Total \neq 2^{w_3}$				
Tip operacije	STARI DEKODER	NOVI DEKODER $r = V \cdot 2^l$		
		$V = 1$	$V = 3$ $V = 5$	$V \geq 7$
Množenje	2	0	1	3
Deljenje	2	1	1	1

Tabela 2.7. Broj operacija množenja i deljenja po dekodovanom simbolu za $Total = 2^{w_3}$

$Total = 2^{w_3}$				
Tip operacije	STARI DEKODER	NOVI DEKODER $r = V \cdot 2^l$		
		$V = 1$	$V = 3$ $V = 5$	$V \geq 7$
Množenje	2	0	1	3
Deljenje	1	0	0	0

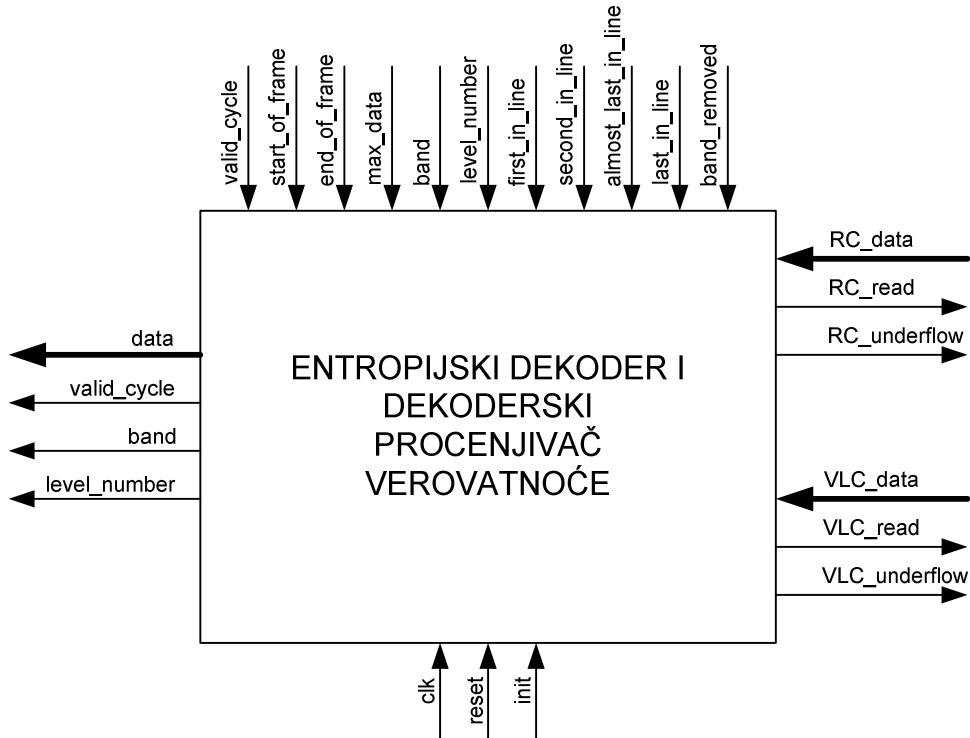
2.3. Hardverska realizacija entropijskog dekodera i dekoderskog procenjivača verovatnoće

Na slici 2.11 je prikazan najviši hijerarhijski nivo hardverske realizacije entropijskog dekodera i dekoderskog procenjivača verovatnoće.

Ulazni podaci za hardversku realizaciju entropijskog dekodera i dekoderskog procenjivača verovatnoće su:

- *RC_data*: podaci koji predstavljaju komprimovanu video sekvencu, kodovani pomoću enkodera opsega; reč je o podacima koji se na ulaz bloka koji predstavlja entropijski dekodera i dekoderski procenjivač verovatnoće dovode za svaki nivo kompozicije ponaosob;
- *VLC_data*: podaci koji predstavljaju komprimovanu video sekvencu, kodovani pomoću VLC enkodera (enkodera sa promenljivom dužinom koda); reč je o podacima koji se na ulaz bloka koji predstavlja entropijski dekodera i

dekoderski procenjivač verovatnoće dovode za svaki nivo kompozicije ponaosob.



Slika 2.11. Najviši hijerarhijski nivo hardverske realizacije entropijskog dekodera i dekoderskog procenjivača verovatnoće

Kontrolni signali i podaci koje generiše blok “SEKVENCER” (koji će biti opisan u šestom poglavlju), a koji se dovode na ulaz bloka koji predstavlja entropijski dekodera i dekoderski procenjivač verovatnoće su:

- *valid_cycle*: kontrolni signal koji kvalifikuje ulazne podatke kao validne ili nevalidne;
- *start_of_frame*: kontrolni signal koji označava početni podatak na nivou frejma za svaki nivo ponaosob;
- *end_of_frame*: kontrolni signal koji označava zadnji podatak na nivou frejma za sve nivoe zajedno;

- *max_data*: kontrolni signal koji označava da li se u trenutnom ciklusu dekoduje maksimum na sledeće četiri linije podataka za svaki nivo i za svaki podopseg ponaosob (koji potom koristi VLC dekodler);
- *band*: podatak koji označava podopseg u okviru tekućeg nivoa kompozicije kojem pripada ulazni podatak;
- *level_number*: podatak koji označava nivo kompozicije kojem pripada ulazni podatak;
- *first_in_line*: kontrolni signal koji označava prvi podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob;
- *second_in_line*: kontrolni signal koji označava drugi podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob;
- *almost_last_in_line*: kontrolni signal koji označava predposlednji podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob;
- *last_in_line*: kontrolni signal koji označava poslednji podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob.

Osim kontrolnih signala i podataka koje generiše blok “SEKVENCER”, i sledeći kontrolni signali i podaci se dovode na ulaz bloka koji predstavlja entropijski dekodler i dekoderski procenjivač verovatnoće:

- *init*: kontrolni signal kojim se pokreće postupak inicijalizacije dekodera;
- *reset*: kontrolni signal za resetovanje dekodera;
- *clk*: signal takta;
- *band_removed*: kontrolni parametar koji određuje da li su i koji podopsezi uklonjeni iz procesa dekompresije.

Izlazni podaci bloka koji predstavlja entropijski dekodler i dekoderski procenjivač verovatnoće su:

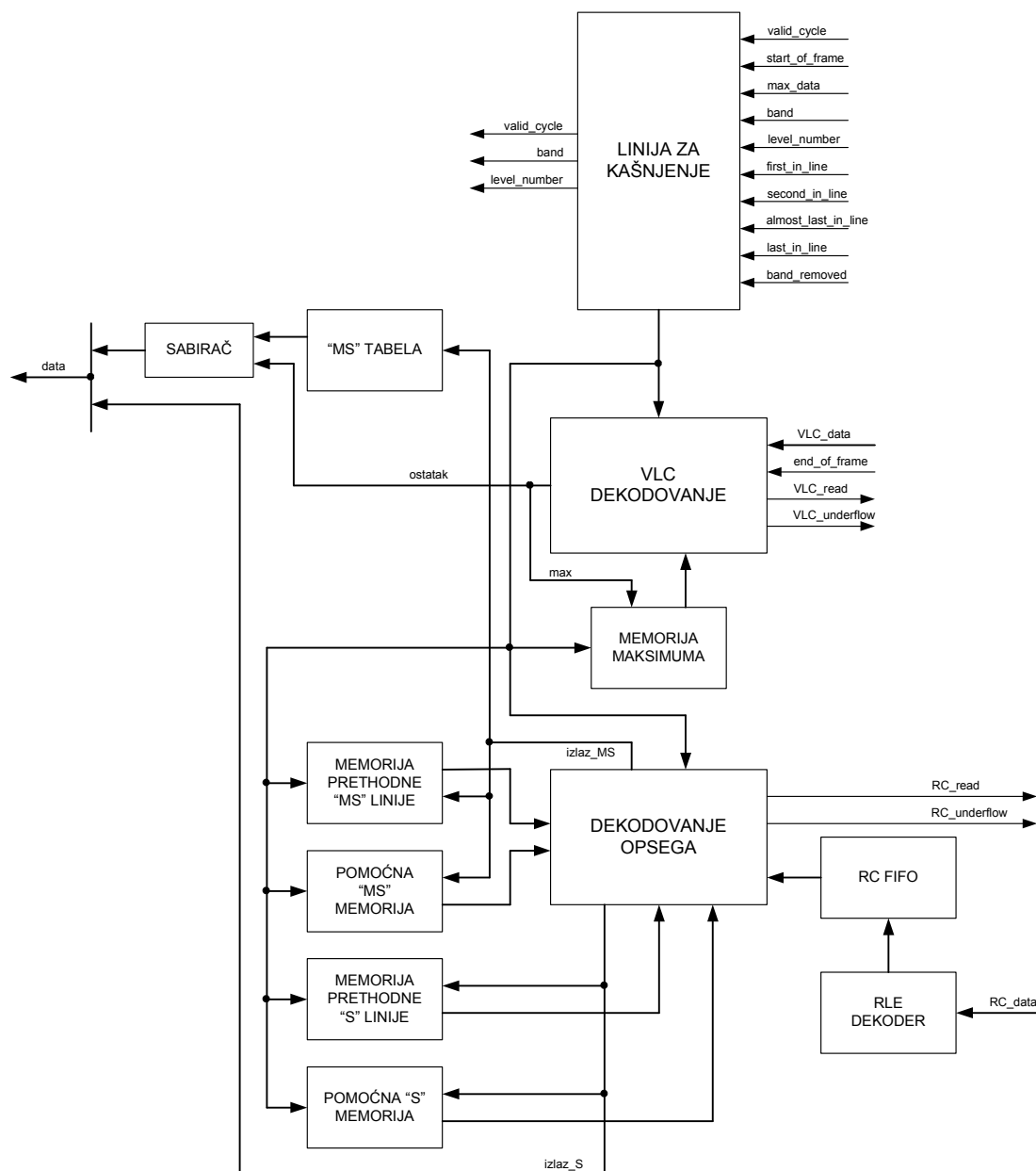
- *data*: dekodovana vrednost odbirka komponente dekomponovanog signala koja se prosleđuje dekvantizatoru;
- *band*: podatak koji označava podopseg u okviru tekućeg nivoa kompozicije kojem pripada izlazni podatak koji se prosleđuje dekvantizatoru;

- *level_number*: podatak koji označava nivo kompozicije kojem pripada izlazni podatak koji se prosleđuje dekvantizatoru.

Osim navedenih izlaznih podataka, postoji još i niz izlaznih kontrolnih signala:

- *valid_cycle*: kontrolni signal koji ukazuje na validnost izlaznog podatka *data* u tekućem ciklusu takta;
- *RC_underflow*: kontrolni signal (za svaki nivo kompozicije ponaosob) koji ukazuje da je narušena dinamika procesa dekompresije usled neblagovremenog učitavanja potrebnih podataka preko ulaza *RC_data*;
- *RC_read*: kontrolni signal (za svaki nivo kompozicije ponaosob) kojim se kontroliše učitavanje potrebnih podataka preko ulaza *RC_data*;
- *VLC_underflow*: kontrolni signal (za svaki nivo kompozicije ponaosob) koji ukazuje da je narušena dinamika procesa dekompresije usled neblagovremenog učitavanja potrebnih podataka preko ulaza *VLC_data*;
- *VLC_read*: kontrolni signal (za svaki nivo kompozicije ponaosob) kojim se kontroliše učitavanje potrebnih podataka preko ulaza *VLC_data*.

Interna struktura bloka koji predstavlja entropijski dekodier i dekoderski procenjivač verovatnoće je prikazana na slici 2.12. Blok “VLC DEKODOVANJE” prihvata podatke kodovane pomoću VLC enkodera koji predstavljaju komprimovanu video sekvencu, vrši njihovo dekodovanje tehnikom promenljive dužine reči i na svom izlazu generiše podatak *ostatak*, koji zapravo predstavlja dekodovani ostatak R (u skladu sa oznakom korišćenom u sekciji 2.2 ove disertacije) koji će biti iskorišćen za izračunavanje vrednosti odbirka komponente dekomponovanog signala *data* (tj. C u skladu sa oznakom korišćenom u sekciji 2.2 ove disertacije). Na istom izlazu blok “VLC DEKODOVANJE” u određenim ciklusa takta (koji su određeni aktivnom vrednošću kontrolnog signala *max_data*) generiše i podatak *max* koji predstavlja maksimum na sledeće četiri linije podataka za odgovarajući nivo kompozicije i za odgovarajući podopseg, koji se potom upisuje u blok “MEMORIJA MAKSIMUMA”.



Slika 2.12. Interna struktura entropijskog dekodera i dekoderskog procenjivača verovatnoće

Blok "VLC DEKODOVANJE" takođe generiše i kontrolne signale *VLC_underflow* (koji ukazuje da je narušena dinamika procesa dekompresije usled neblagovremenog učitavanja potrebnih podataka preko ulaza *VLC_data*, što onemogućava dalji nastavak procesa dekodovanja i zahteva resetovanje celog procesa) i *VLC_read* (kojim se kontroliše učitavanje potrebnih podataka preko ulaza *VLC_data*).

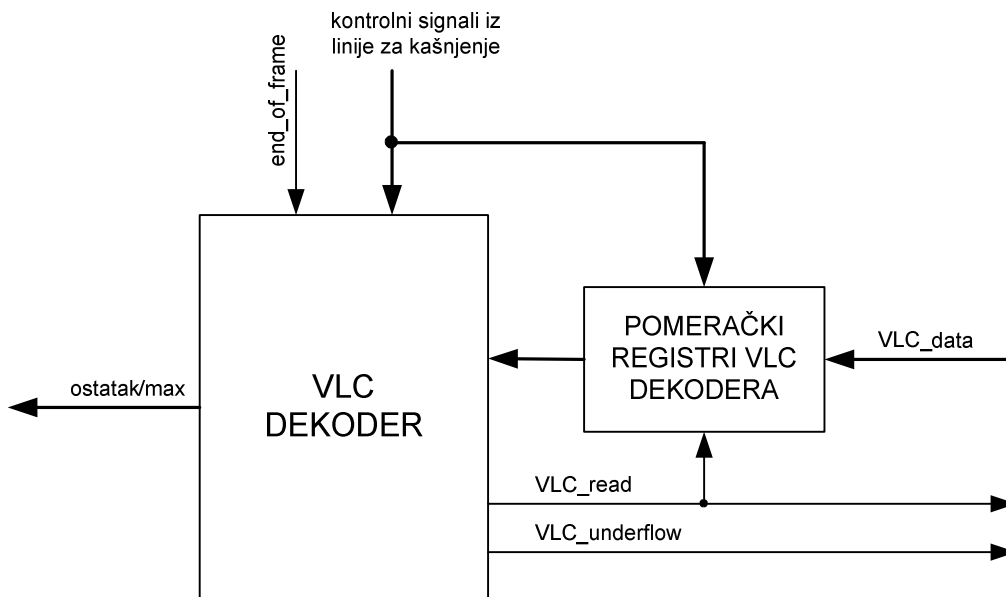
Blok “RLE DEKODER” prihvata podatke kodovane pomoću enkodera opsega koji predstavljaju komprimovanu video sekvencu, vrši njihovo dekodovanje tehnikom *run-length* dekodovanja i upisuje ih u FIFO memoriju “RC FIFO”. Blok “DEKODOVANJE OPSEGA” očitava podatke iz FIFO memorije “RC FIFO”, vrši njihovo dekodovanje i na svom izlazu generiše podatke *izlaz_MS* i *izlaz_S*. Podatak *izlaz_MS* predstavlja indeks amplitudne grupe koji se dovodi na ulaz bloka “MS TABELA”, koji u skladu sa tabelom 2.3 određuje donju granicu odbiraka komponenti dekomponovanog signala, koja se potom u bloku “SABIRAČ” sabira sa dekodovanim ostatkom R u cilju formiranja dekodovane vrednosti amplitude M . Podatak *izlaz_S* predstavlja dekodovanu vrednost znaka S . Podatak sa izlaza bloka “SABIRAČ” i podatak *izlaz_S* zajedno predstavljaju vrednost odbirka komponente dekomponovanog signala *data* (tj. C u skladu sa oznakom korišćenom u sekciji 2.2 ove disertacije).

Podaci *izlaz_MS* se upisuju i u blokove “MEMORIJA PRETHODNE MS LINIJE” (koja sadrži već dekodovane MS podatke iz prethodne linije) i “POMOĆNA MS MEMORIJA” (u kojoj se čuva susedni levi dekodovani MS podatak iz iste linije u odnosu na podatak koji se trenutno dekoduje). Čuvanje ovih podataka je neophodno da bi se u procesu dekodovanja indeksa amplitudne grupe MS izračunala srednja vrednost susednih indeksa amplitudne grupe \overline{MS} (prikazanih na slici 2.2) potrebna za određivanje konteksta amplitude MC , kao što je to opisano u sekciji 2.2. Podaci *izlaz_S* se upisuju i u blokove “MEMORIJA PRETHODNE S LINIJE” (koja sadrži već dekodovane S podatke iz prethodne linije) i “POMOĆNA S MEMORIJA” (u kojoj se čuva susedni levi dekodovani S podatak iz iste linije u odnosu na podatak koji se trenutno dekoduje). Čuvanje ovih podataka (susednih vrednosti znakova S_i prikazanih na slici 2.3) je neophodno da bi se na osnovu njih dekodovao ternarni kontekst TC kao što je to opisano u sekciji 2.2.

Blok “DEKODOVANJE OPSEGA” takođe generiše i kontrolne signale *RC_underflow* (koji ukazuje da je narušena dinamika procesa dekompresije usled neblagovremenog učitavanja potrebnih podataka preko ulaza *RC_data*, što onemogućava dalji nastavak procesa dekodovanja i zahteva resetovanje celog procesa) i *RC_read* (kojim se kontroliše učitavanje potrebnih podataka preko ulaza *RC_data*).

Kontrolni signali i podaci koji se dovode na ulaz bloka “LINIJA ZA KAŠNJENJE” dalje se prosleđuju, zakašnjeni za odgovarajući broj perioda takta, i do

ostalnih blokova, kao što je to predstavljeno na slici 2.12. Pri tome, kontrolni signali i podaci *valid_cycle*, *band* i *level_number* se nakon prolaska kroz liniju za kašnjenje, prosleđuju i na izlaz entropijskog dekodera i dekoderskog procenjivača verovatnoće. Blok “LINIJA ZA KAŠNENJE” obezbeđuje ispravnu protočnu obradu podataka čime se povećava maksimalna radna frekvencija dizajna.

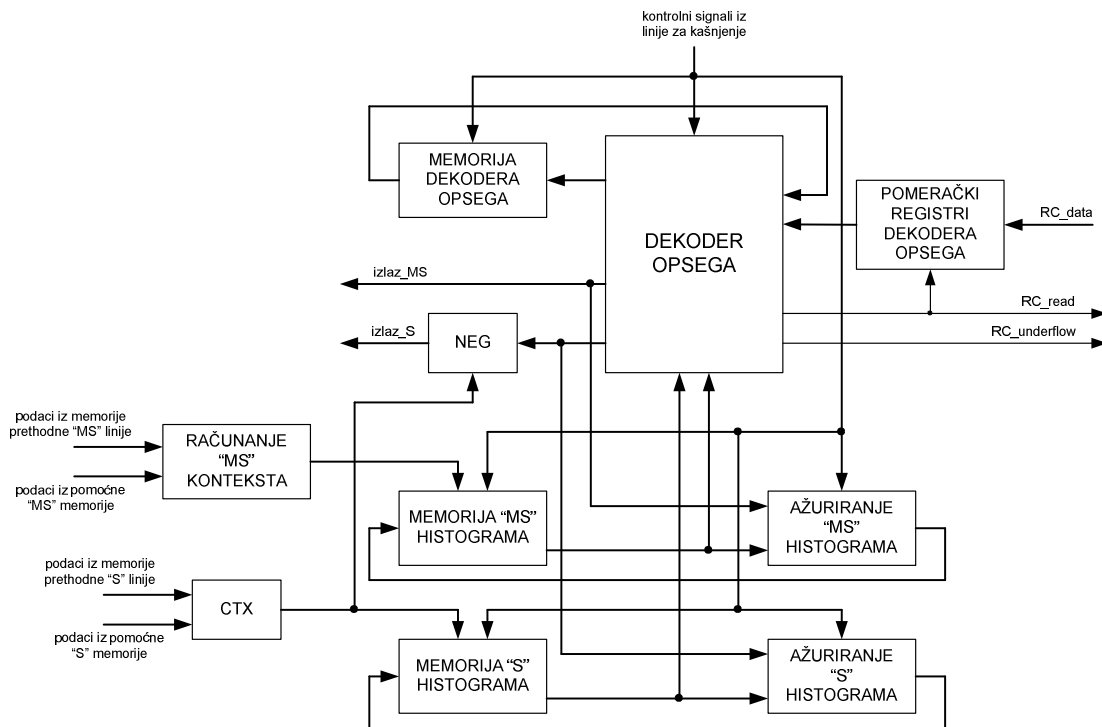


Slika 2.13. Interna struktura bloka “VLC DEKODOVANJE”

Interna struktura bloka “VLC DEKODOVANJE” je prikazana na slici 2.13. Sam proces dekodovanja tehnikom promenljive dužine reči se vrši u bloku “VLC DEKODER”, dok se u bloku “POMERAČKI REGISTRI VLC DEKODERA” nalaze biti kojima predstoji proces dekodovanja. Blok “POMERAČKI REGISTRI VLC DEKODERA” sadrži ukupno sedam 16-bitnih registara (za svaki nivo kompozicije po jedan registar) sa podacima za VLC dekodovanje, i još sedam registara koji sadrže informaciju o broju validnih bita u pomenutim registrima sa podacima za VLC dekodovanje.

Interna struktura bloka “DEKODOVANJE OPSEGA” je prikazana na slici 2.14. Ulazni enkodovani podaci *RC_data* se u bloku “POMERAČKI REGISTRI DEKODERA OPSEGA” pakuju u pakete od po tri bajta, koji se potom paralelno

prosleđuju bloku “DEKODER OPSEGA” (i to nula, jedan, dva ili tri bajta paralelno), u kojem se zatim vrši proces dekodovanja opsega.



Slika 2.14. Interna struktura bloka “DEKODOVANJE OPSEGA”

U bloku “MEMORIJA DEKODERA OPSEGA” se čuvaju parametri R , B i d opisani u sekciji 2.2 ove disertacije. Blok “RAČUNANJE MS KONTEKSTA” izračunava kontekst indeksa amplitudne grupe na osnovu četiri susedna već dekodovana MS podatka, tj. određuje koji MS histogram treba da se koristi u dekodovanju tekućeg MS podatka (indeksa amplitudne grupe). Odgovarajući MS histogram se potom očitava iz bloka “MEMORIJA MS HISTOGRAMA”, prosleđuje se na ulaz bloka “DEKODER OPSEGA” kako bi bio iskorišćen u procesu dekodovanja tekućeg MS podatka, i istovremeno se prosleđuje na ulaz bloka “AŽURIRANJE MS HISTOGRAMA” kako bi se sproveo proces ažuriranja MS histograma. Nakon završetka procesa ažuriranja, ažurirani MS histogram se upisuje u blok “MEMORIJA MS HISTOGRAMA”. Blok “CTX” izračunava kontekst znaka na osnovu četiri susedna već dekodovana S podatka, tj. određuje koji S histogram treba da se koristi u dekodovanju tekućeg S podatka. Odgovarajući S histogram se potom očitava iz bloka “MEMORIJA S HISTOGRAMA”,

prosleđuje se na ulaz bloka “DEKODER OPSEGA” kako bi bio iskorišćen u procesu dekodovanja tekućeg S podatka, i istovremeno se prosleđuje na ulaz bloka “AŽURIRANJE S HISTOGRAMA” kako bi se sproveo proces ažuriranja S histograma. Nakon završetka procesa ažuriranja, ažurirani S histogram se upisuje u blok “MEMORIJA S HISTOGRAMA”. U bloku “NEG” se vrši ispitivanje znaka S i, ako je potrebno, inverzija znaka S u skladu sa tabelom 2.2, kao što je to opisano u sekciji 2.2.

Kontrolni signali i podaci iz bloka “LINIJA ZA KAŠNJENJE” se dovode do odgovarajućih blokova, kao što je to predstavljeno na slici 2.14, obezbeđujući ispravnu protočnu obradu podataka čime se povećava maksimalna radna frekvencija dizajna.

3. Dekvantizator

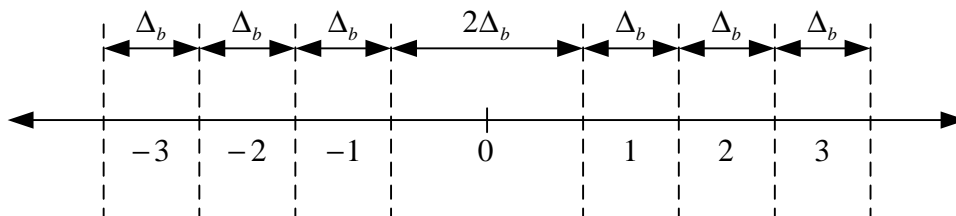
3.1. Uvod

Dekvantizacija odbiraka koji predstavljaju komponente dekomponovanog signala se obavlja u dekvantizatoru samo u slučaju kompresije sa gubicima. U slučaju kompresije bez gubitaka, komponente dekomponovanog signala se direktno propuštaju kroz dekvantizator i dovode na ulaz inverznog podopsežnog transformatora. S obzirom da brzi dekoder komprimovane slike, opisan u ovoj disertaciji, treba da bude kompatibilan sa već razvijenim hardverskim enkoderom [24], čija je realizacija opisana u [30], potrebno je i da dekvantizator realizovan u ovoj disertaciji bude kompatibilan za kvantizatorom realizovanim u okviru hardverskog enkodera [30].

Zarad boljeg razumevanja procesa dekvantizacije koji obavlja realizovani dekvantizator, najpre će ukratko biti opisan postupak kvantizacije koji se primenjuje u realizaciji hardverskog enkodera [30]. U toj realizaciji je usvojena implementacija uniformnog skalarnog kvantizatora sa mrtvom zonom [31] koja se koristi i u JPEG 2000 standardu [32]. Kvantizacija se vrši pomoću uniformnog skalarnog kvantizatora sa mrtvom zonom, čiji je korak kvantizacije Δ_b , a širina mrtve zone $2\Delta_b$, kao što je to prikazano na slici 3.1. U opštem slučaju, za različite podopsege se koriste različiti koraci kvantizacije. Korak kvantizacije Δ_b , za podopseg b , se izračunava na osnovu dinamičkog opsega vrednosti odbiraka koji predstavljaju komponente dekomponovanog signala u podopsegu b . S obzirom da odbirci koji predstavljaju komponente dekomponovanog signala nemaju isti dinamički opseg na svim podopsezima nakon dekompozicije, kvantizacijom pomoću različitih vrednosti koraka kvantizacije se postiže bolji stepen kompresije. Rad uniformnog skalarnog kvantizatora sa mrtvom zonom je opisan sledećom jednačinom:

$$q_b = \text{sign}(y_b) \cdot \left\lfloor \frac{|y_b|}{\Delta_b} \right\rfloor, \quad (3.1)$$

gde y_b predstavlja odbirak komponente dekomponovanog signala iz podopsega b , Δ_b predstavlja korak kvantizacije za podopseg b , a q_b predstavlja rezultujuću kvantizovanu vrednost odbirka komponente dekomponovanog signala. Prilikom izračunavanja količnika u izrazu (3.1), vrši se odsecanje sa donje strane.



Slika 3.1. Ilustracija rada uniformnog skalarnog kvantizatora sa mrtvom zonom u JPEG 2000 standardu

Kako bi se izbegla operacija deljenja, i time redukovala složenost hardverske realizacije kvantizatora, za korak kvantizacije Δ_b su usvojene vrednosti koje predstavljaju celobrojni umnožak stepena broja 2. Konkretno, za podopsege LH i HL na nivou i dekompozicije, korak kvantizacije je:

$$\Delta_{LHi,HLi} = M \cdot 2^{E-i}. \quad (3.2)$$

S obzirom da podopsezi HHi imaju veći dinamički opseg od podopsega LHi i HLi, za podopsege HHi je usvojen i veći korak kvantizacije:

$$\Delta_{HHi} = M \cdot 2^{E-i+1}. \quad (3.3)$$

Imajući u vidu da podopsezi LLi imaju manji dinamički opseg od podopsega LHi i HLi, za podopsege LLi je usvojen i manji korak kvantizacije:

$$\Delta_{LLi} = M \cdot 2^{E-i-1}. \quad (3.4)$$

Parametar M predstavlja mantisu osnovnog koraka kvantizacije, i može imati vrednosti iz opsega $64 \leq M \leq 127$, dok parametar E predstavlja eksponent osnovnog koraka kvantizacije, i može imati vrednosti iz opsega $-6 \leq E \leq 6$. Konačno, jednačine koje opisuju rad kvantizatora iz [30] za podopsege LHi, HLi i HHi, su:

$$q_{LHi} = \text{sign}(y_{LHi}) \cdot \left\lfloor \frac{|y_{LHi}|}{M \cdot 2^{E-i}} \right\rfloor = \text{sign}(y_{LHi}) \cdot \left\lfloor \frac{|y_{LHi}|}{M} \right\rfloor \cdot 2^{-(E-i)}, \quad (3.5)$$

$$q_{HLi} = \text{sign}(y_{HLi}) \cdot \left\lfloor \frac{|y_{HLi}|}{M \cdot 2^{E-i}} \right\rfloor = \text{sign}(y_{HLi}) \cdot \left\lfloor \frac{|y_{HLi}|}{M} \right\rfloor \cdot 2^{-(E-i)}, \quad (3.6)$$

$$q_{HHi} = \text{sign}(y_{HHi}) \cdot \left\lfloor \frac{|y_{HHi}|}{M \cdot 2^{E-i+1}} \right\rfloor = \text{sign}(y_{HHi}) \cdot \left\lfloor \frac{|y_{HHi}|}{M} \right\rfloor \cdot 2^{-(E-i+1)}. \quad (3.7)$$

S obzirom da se u podopsezima LLi nalazi veoma bitan sadržaj, prilikom kvantizacije tih podopsega se ne vrši odsecanje već zaokruživanje [33]:

$$q_{LLi} = \text{sign}(y_{LLi}) \cdot \left\lfloor \frac{2|y_{LLi}| + M \cdot 2^{E-i-1}}{M \cdot 2^{E-i}} \right\rfloor = \text{sign}(y_{LLi}) \cdot \left\lfloor \frac{2|y_{LLi}| + M \cdot 2^{E-i-1}}{M} \right\rfloor \cdot 2^{-(E-i)}. \quad (3.8)$$

Dekvantizator realizovan u sklopu hardverske realizacije brzog dekodera opisanog u ovoj disertaciji, vrši postupak dekvantizacije odbiraka koji predstavljaju komponente dekomponovanog signala postupkom koji je u potpunosti kompatibilan, ali inverzan opisanom postupku kvantizacije.

Dekvantizovane vrednosti modula (apsolutne vrednosti) odbiraka koji predstavljaju komponente dekomponovanog signala iz podopsega HHi, LHi i HLi, se izračunavaju korišćenjem postupka inverznog postupku zaokruživanja:

$$|y_{LHi_deq}| = |q_{LHi}| \cdot M \cdot 2^{E-i} + M \cdot 2^{E-i-1}, \quad (3.9)$$

$$|y_{HLi_deq}| = |q_{HLi}| \cdot M \cdot 2^{E-i} + M \cdot 2^{E-i-1}, \quad (3.10)$$

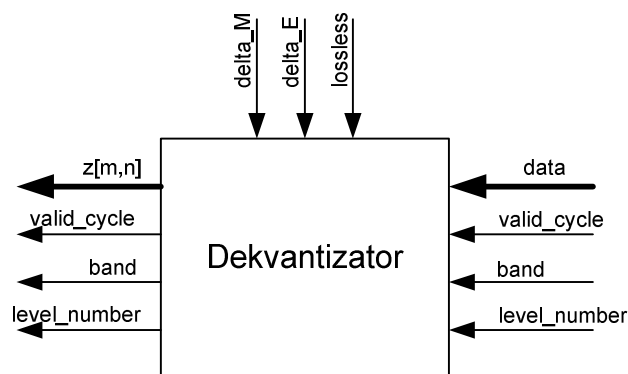
$$|y_{HHi_deq}| = |q_{HHi}| \cdot M \cdot 2^{E-i+1} + M \cdot 2^{E-i}. \quad (3.11)$$

Dekvantizovane vrednosti modula (apsolutne vrednosti) odbiraka koji predstavljaju komponente dekomponovanog signala iz podopsega LLi se izračunavaju korišćenjem postupka inverznog postupku odsecanja:

$$|y_{LLi_deq}| = |q_{LLi}| \cdot M \cdot 2^{E-i-1}. \quad (3.12)$$

3.2. Hardverska realizacija dekvantizatora

Na slici 3.2 je prikazan najviši hijerarhijski nivo hardverske realizacije dekvantizatora.



Slika 3.2. Najviši hijerarhijski nivo hardverske realizacije dekvantizatora

Njegovi ulazni podaci i signali su:

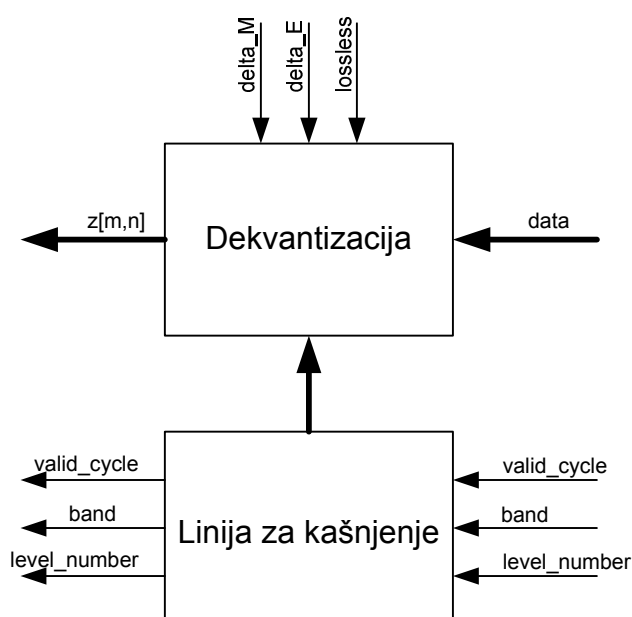
- *data*: podaci koji predstavljaju kvantizovanu vrednost odbirka komponente dekomponovanog signala koja se prosleđuje dekvantizatoru od strane entropijskog dekodera i dekoderskog procenjivača verovatnoće;
- *valid_cycle*: kontrolni signal koji kvalifikuje ulazne podatke *data* u tekućem ciklusu takta kao validne ili nevalidne;
- *band*: kontrolni podatak koji označava podopseg u okviru tekućeg nivoa kompozicije kojem pripada ulazni podatak *data* u tekućem ciklusu takta;
- *level_number*: kontrolni podatak koji označava nivo kompozicije kojem pripada ulazni podatak *data* u tekućem ciklusu takta;
- *delta_M*: kontrolni parametar koji definiše mantisu za osnovni korak dekvantizacije;
- *delta_E*: kontrolni parametar koji definiše eksponent za osnovni korak dekvantizacije;
- *lossless*: kontrolni parametar koji određuje da li se dekompresija video sekvence vrši sa ili bez gubitaka.

Izlazni podaci i signali dekvantizatora su:

- $z[m,n]$: podaci koji predstavljaju dekvantizovanu vrednost odbirka komponente dekomponovanog signala;
- *valid_cycle*: izlazni kontrolni signal koji kvalifikuje izlazne podatke $z[m,n]$ u tekućem ciklusu takta kao validne ili nevalidne;
- *band*: izlazni podatak koji označava podopseg u okviru tekućeg nivoa kompozicije kojem pripada izlazni podatak $z[m,n]$ u tekućem ciklusu takta;
- *level_number*: izlazni podatak koji označava nivo kompozicije kojem pripada izlazni podatak $z[m,n]$ u tekućem ciklusu takta.

Interna struktura hardverske realizacije dekvantizatora je prikazana na slici 3.3. Blok “DEKVANTIZACIJA” vrši dekvantizaciju ulaznog podatka *data* koji predstavlja kvantizovanu vrednost odbirka komponente dekomponovanog signala u formatu “znak plus apsolutna vrednost”. Dekvantizacija apsolutne vrednosti ulaznog podatka se vrši na način opisan u prethodnoj sekciji uz zadržavanje istog znaka podatka, nakon čega se

vrši konverzija koda dekvantizovanog odbirka iz formata “znak plus apsolutna vrednost” u binarni format u kojem se negativne vrednosti predstavljaju u komplementu dvojke, čime se kompletira proces generisanja izlaznog podatka $z[m,n]$. Kontrolni signali i podaci iz bloka “LINIJA ZA KAŠNENJE” se dovode do bloka “DEKVANTIZACIJA”, da bi obezbedili ispravnu protočnu obradu podataka sa ciljem da se poveća maksimalna radna frekvencija dizajna. Zakašnjena replika tih kontrolnih signala se takođe prosleđuje i na izlaz, kako bi se obezbedilo dalje ispravno procesiranje podataka $z[m,n]$ u inverznom podopsežnom transformatoru.



Slika 3.3. Interna struktura hardverske realizacije dekvantizatora

4. Inverzni podopsežni transformator

4.1. Uvod

S obzirom da se prilikom kompresije digitalne slike najčešće vrši nekakav oblik razlaganja slike pre postupka kodovanja entropijskim koderom, jer se u protivnom ne bi mogao ostvariti zadovoljavajući stepen kompresije, i kod dekompresije digitalne slike je neophodno da postoji proces odgovarajuće kompozicije sastavnih delova u sliku kao celinu. Postupci koji se koriste za razlaganje slike prilikom postupka kompresije i kompoziciju slike prilikom postupka dekompresije su predviđajući postupci, blokovske transformacije i podopsežne transformacije. Predviđajući postupci se najčešće koriste za kompresiju/dekompresiju slika bez gubitaka i sa malim stepenom kompresije. Blokovski transformacioni postupci se često koriste (među njima je jedan od najpoznatijih postupaka diskretna kosinusna transformacija), ali njihov najveći nedostatak su blokovski artefakti, na koje je ljudski vizuelni sistem veoma osetljiv, što se najlakše primećuje pri većim stepenima kompresije i u područjima slike sa malom lokalnom varijansom. Podopsežni transformacioni postupci se mogu koristiti i za kompresiju/dekompresiju slika bez gubitaka i za kompresiju/dekompresiju slika sa gubicima, pri čemu je jedini vidljivi artefakt koji se javlja tzv. efekat talasanja, objašnjen u [34], koji je posledica Gibsovog fenomena kod linearnih filtara. Zbog svojih dobrih osobina, podopsežni transformacioni postupci se koriste i u poznatom standardu za kompresiju mirne slike JPEG 2000 [32]. Za kompresiju/dekompresiju slika sa gubicima se u JPEG 2000 standardu koristi dvodimenzionalna diskretna *wavelet* transformacija (DWT).

Prednosti DWT u odnosu na ostale transformacione postupke su opisane u [35]. Diskretna *wavelet* transformacija je postala standardna tehnika u procesiranju audio signala i kompresiji/dekompresiji digitalne slike otkada je u [36] predložena

višerezolucijska predstava signala zasnovana na *wavelet* dekompoziciji. Prema JPEG 2000 standardu prilikom kompresije slike, nekomprimovana slika ili njeni delovi se dekomponuju u različite nivoe dekompozicije korišćenjem dvodimenzionalne DWT. Svaki od nivoa dekompozicije se sastoji od podopsega koji sadrže odbirke signala nastale filtriranjem, koji opisuju horizontalne i vertikalne karakteristike prostorne frekvencije nekomprimovane slike ili njenih delova. U svrhu kreiranja spektralnih komponenti signala se koriste jednodimenzionalni filtri, koji na svom izlazu generišu niskofrekventne i visokofrekventne komponente signala, horizontalno duž linija i vertikalno duž kolona slike ili njenih delova. Sličan, ali inverzan postupak se primenjuje i kod procesa dekompresije digitalne slike.

Kao što je pokazano u [37-39], izbor filtera je veoma značajno pitanje, s obzirom da filtri predstavljaju važan faktor koji utiče na performanse procesa dekompozicije (kompozicije) pri kompresiji (dekompresiji) digitalne slike. Prema JPEG 2000 standardu, za potrebe kompresije/dekompresije digitalne slike bez gubitaka se koriste *Le Gall*-ovi 5/3 filtri [40], koji omogućavaju savršenu rekonstrukciju signala.

Diferencne jednačine (4.1) i (4.2) opisuju *Le Gall*-ov niskopropusni i visokopropusni jednodimenzionalni (1-D) direktni DWT 5/3 filter, respektivno, pri čemu $x[n]$ predstavlja odbirke ulaznog signala, $y_0[n]$ predstavlja odbirke izlaznog signala koji su generisani niskopropusnim direktnim filtriranjem ulaznih odbiraka, a $y_1[n]$ predstavlja odbirke izlaznog signala koji su generisani visokopropusnim direktnim filtriranjem ulaznih odbiraka.

$$y_0[n] = -\frac{1}{8}x[n] + \frac{1}{4}x[n-1] + \frac{3}{4}x[n-2] + \frac{1}{4}x[n-3] - \frac{1}{8}x[n-4], \quad (4.1)$$

$$y_1[n] = -\frac{1}{2}x[n-1] + x[n-2] - \frac{1}{2}x[n-3]. \quad (4.2)$$

Diferencne jednačine (4.3) i (4.4) opisuju *Le Gall*-ov niskopropusni i visokopropusni 1-D inverzni DWT 5/3 filter, respektivno, pri čemu $w_0[n]$ predstavlja odbirke izlaznog signala koji su generisani niskopropusnim inverznim filtriranjem odbiraka $y_0[n]$, dok $w_1[n]$ predstavlja odbirke izlaznog signala koji su generisani visokopropusnim inverznim filtriranjem odbiraka $y_1[n]$.

$$w_0[n] = \frac{1}{2}y_0[n-1] + y_0[n-2] + \frac{1}{2}y_0[n-3], \quad (4.3)$$

$$w_1[n] = -\frac{1}{8}y_1[n] - \frac{1}{4}y_1[n-1] + \frac{3}{4}y_1[n-2] - \frac{1}{4}y_1[n-3] - \frac{1}{8}y_1[n-4]. \quad (4.4)$$

Kada je reč o hardverskim realizacijama filtara, JPEG 2000 standard podržava realizacije zasnovane na konvoluciji i realizacije zasnovane na dizanju (*lifting*). Hardverske realizacije DWT zasnovane na konvoluciji uglavnom zahtevaju izvršavanje većeg broja aritmetičkih operacija i povećani utrošak logičkih i memorijskih resursa, što nije poželjno u primenama koje zahtevaju veliku brzinu rada i malu potrošnju. Hardverske realizacije 5/3 DWT, zasnovane na konvoluciji su opisane u [41-46]. Hardverske realizacije DWT zasnovane na dizanju imaju mnoge prednosti u poređenju sa realizacijama zasnovanim na konvoluciji koje se ogledaju pre svega u jednostavnijem dizajnu, manjem utrošku logičkih i memorijskih resursa, manjoj složenosti računskih operacija i nižoj potrošnji. U literaturi su predložene brojne hardverske realizacije 5/3 DWT zasnovane na dizanju.

Dizajn filtra na bazi direktnog mapiranja je predložen u [47,48]. Međutim, za odgovarajući dizajn 5/3 filtra, u situaciji kada se parni i neparni odbirci signala čitaju sekvencijalno i baferišu, ceo dizajn radi 50% sporije. Ovaj dizajn je unapređen savijanjem poslednja dva *pipeline* stepena i povezivanjem sa prva dva stepena u [49]. Za odgovarajući dizajn 5/3 filtra, s obzirom da se sastoji od samo dva *pipeline* stepena, pomenuto savijanje se ne može izvršiti. Generalisani dizajn filtra, koji je predložen u [50], predstavlja primer programabilnog dizajna koji podržava realizacije većeg broja filtara, uključujući i 5/3 filter. Iako neki konvencionalni dizajni filtra, zasnovani na dizanju, zahtevaju manji broj aritmetičkih operacija, oni ponekad imaju dužu kritičnu putanju. Rešavanje ovog problema dodavanjem *pipeline* registara bi rezultovalo značajnijim povećanjem broja korišćenih registara. U [51] je opisano rešenje koje problem akumulacije kašnjenja rešava na efikasan način. Programabilni filter čiji je dizajn zasnovan na množenju i akumulaciji međurezultata je predložen u [52]. Za razliku od većine tradicionalnih realizacija filtara za DWT, koji izračunavaju naredni nivo dekompozicije/kompozicije tek nakon kompletiranja izračunavanja prethodnog nivoa, rekurzivni dizajn DWT filtra prezentovan u [53] je sposoban da procesira više

nivoa dekompozicije/kompozicije istovremeno. Dizajn filtra sa dualnim skeniranjem za DWT, koji procesira dva nezavisna niza ulaznih podataka, učešljanim korišćenjem istih funkcionalnih blokova, je opisan u [54]. Dizajn u vidu digitalnog procesora signala, koji može biti programiran da podrži veći broj različitih tipova filtara, uključujući i 5/3 filtre, je predložen u [55]. Optimizovani, na sabiračima zasnovani, dizajn 5/3 filtra, pogodan za implementaciju sa malom potrošnjom i malom površinom dizajna, je prezentovan u [56].

Kao što je pokazano u [57], u pogledu hardverske kompleksnosti dizajn [49] je najjednostavniji, dok je dizajn [55] najslabiji. Dizajni filtara opisani u [47], [50-54] i [56] imaju sličnu hardversku kompleksnost i razlikuju se uglavnom u broju registara i sabirača. Kompleksnost upravljanja filtrom [47] je veoma jednostavna, za razliku od kompleksnosti upravljanja filtrima [54,55] koja je prilično složena. Ostali pomenuti dizajni filtra imaju osrednju kompleksnost upravljanja. U pogledu brzine protoka podataka, dizajni [47], [49] i [50-52] su najbolji. Dizajn [53] ima kraće vreme procesiranja podataka, ali po cenu niže maksimalne frekvencije na kojoj može da radi, dok dizajn filtra [51] ima najmanje kašnjenje. Sve realizacije osim [53] izračunavaju sve odbirke signala na jednom nivou dekompozicije/kompozicije pre nego što počnu sa izračunavanjima na sledećem nivou, dok jedino realizacija [53] učešljava izračunavanja odbiraka signala sa viših nivoa dekompozicije/kompozicije sa izračunavanjima na nižim nivoima uključujući i nivo 1. Iz tog razloga je utrošak memorijskih resursa za realizaciju [53] manji u poređenju sa ostalim realizacijama.

Među pomenutim hardverskim realizacijama DWT zasnovanim na dizanju, u pogledu kompleksnosti njihovih osnovnih gradivnih blokova (jednodimenzionalnih 5/3 filtara), najjednostavniji su dizajni [47-50] i [56], koji omogućavaju značajne uštede logičkih i memorijskih resursa. Hardverske realizacije prezentovane u [51-55] imaju veću hardversku kompleksnost, ali i veću fleksibilnost dizajna uključujući i podršku za široki opseg različitih tipova filtara.

Do sada je razvijen i veći broj dvodimenzionalnih (2-D) DWT arhitektura, baziranih na *Le Gall*-ovim 5/3 filtrima.

Neseparabilna 2-D DWT arhitektura i SIMD arhitektura su prezentovane u [58]. Direktna implementacija 2-D DWT (tzv. direktna arhitektura), kao i implementacija koja upotrebljava dva niza filtara i dva paralelna filtra (tzv. sistolno-paralelna

arhitektura) su predložene u [59]. Paralelna 2-D DWT arhitektura, zasnovana na modifikovanom rekurzivnom piramidalnom algoritmu opisanom u [58], je predložena u [60,61]. Arhitektura za 2-D DWT bazirana na sistemu filtriranja "liniju po liniju" sa redukovanim utroškom memorijskih resursa je opisana u [62]. Već pomenuti dizajn filtra [52] je upotrebljen i za realizaciju odgovarajuće programabilne 2-D DWT arhitekture. Veoma brza *pipeline* 2-D DWT arhitektura, zasnovana na konvoluciji, sa niskim nivoom potrošnje, koja vrši transformaciju 4 različita podopsega paralelno, prezentovana je u [63]. Dizajn filtra zasnovan na konvoluciji [42] je iskorišćen za razvoj arhitekture za 2-D DWT bazirane na sistemu filtriranja "liniju po liniju" u kojoj se koristi polifazna dekompozicija i tehnika savijanja sa ciljem povećanja stepena hardverskog iskorišćenja i smanjenja ukupnog vremena procesiranja. Generička 2-D DWT arhitektura, zasnovana na dizanju, koja koristi tehnike paralelovanja uz rekurzivni piramidalni algoritam, je prezentovana u [64]. Ova arhitektura vrši dekompoziciju/kompoziciju u više nivoa korišćenjem metode učešljavanja čime se redukuje kapacitet korišćene memorije i broj pristupa memoriji, ali uz nizak stepen hardverskog iskorišćenja i sporiji protok podataka. Generalisani dizajn filtra [50] je upotrebljen za razvoj blokovski orijentisane implementacije 2-D DWT arhitekture sa četiri procesora, koja ima visok stepen programabilnosti, ali koja zahteva i korišćenje značajnih memorijskih resursa. Rekurzivni dizajn filtra zasnovan na dizanju [53], je iskorišćen za realizaciju 2-D DWT rekurzivne arhitekture (RA) koja procesira različite nivoe dekompozicije/kompozicije istovremeno. Dizajn filtra sa dualnim skeniranjem [54] je upotrebljen u okviru 2-D DWT arhitekture sa dualnim skeniranjem (DSA) koja koristi metod učešljavanja prilikom procesa dekompozicije/kompozicije slike, uz redukovan utrošak memorijskih resursa i smanjen broj pristupa memoriji. Dvodimenzionalna DWT arhitektura koja predstavlja hibrid između realizacije bazirane na sistemu filtriranja "liniju po liniju" i realizacije bazirane na sistemu filtriranja "nivo po nivo" i u kojoj se slika skenira pomoću procesora vrsta, je opisana u [65]. Generička RAM-bazirana 2-D DWT arhitektura visoke efikasnosti i visokog nivoa fleksibilnosti, sa više nivoa dekompozicije/kompozicije, koja koristi sistem filtriranja "liniju po liniju" je opisana u [66]. U ovu arhitekturu se lako mogu integrisati 1-D DWT dizajni filtra zasnovani na konvoluciji kao i dizajni filtra zasnovani na dizanju. Efikasna hardverska realizacija koja koristi *pipeline* tehniku i koja ima veoma kratku kritičnu putanju koja

sadrži samo jedan množač, prezentovana je u [67]. U radu [68] je istražena paralelizacija između 4 transformaciona podopsega u 2-D DWT arhitekturi zasnovanoj na dizanju i predložena je efikasna realizacija bazirana na savijanju (tzv. *folded architecture* – FA). Dvodimenzionalna DWT *pipeline* arhitektura (PA), pogodna za primenu u sistemima sa velikom brzinom rada i malom potrošnjom, je takođe predstavljena u [68]. Brza 2-D DWT arhitektura (tzv. *fast architecture* – FA), sastavljena od dva horizontalna modula za filtriranje i jednog vertikalnog modula za filtriranje, koja koristi *pipeline* tehnike i tehnike paralelne obrade podataka, kao i veoma brza 2-D DWT arhitektura (tzv. *high-speed architecture* – HA), koja koristi paralelizam između četiri transformaciona podopsega, predložene su u [69]. Dvodimenzionalna DWT arhitektura zasnovana na blokovskoj realizaciji, koja eliminiše potrebu za postojanjem *frame buffer*-a, ali koja zato koristi internu memoriju većeg kapaciteta na čipu i ima veći utrošak logičkih resursa zbog postojanja posebnog ulaznog interfejsa, opisana je u [70]. Paralelna 2-D DWT zasnovana na dizanju, koja procesira veći broj nivoa dekompozicije/kompozicije pomoću jednog procesora koji računa i predikcione korake i korake za ažuriranje unutar osnovnog 5/3 filtra zasnovanog na dizanju, predstavljena je u [71]. Memorijski veoma efikasna i brza 2-D DWT arhitektura zasnovana na konvoluciji, koja podržava dekompoziciju/kompoziciju u više nivoa, prezentovana je u [72]. Međutim, ovo rešenje je pogodno samo za 2-D DWT sa manjim brojem nivoa dekompozicije/kompozicije, jer se za veći broj nivoa značajno povećava kompleksnost izračunavanja i kompleksnost hardvera. Dvodimenzionalna DWT arhitektura sa malim utroškom memorijskih resursa, koja vrši kombinovanu obradu podataka duž vrsta i duž kolona slike, opisana je u [73]. Za izračunavanje 2-D DWT u radu [74] su prezentovane savijena arhitektura sa više nivoa dekompozicije/kompozicije (tzv. *folded multilevel architecture* – FMA) i *pipeline* arhitektura sa više nivoa dekompozicije/kompozicije (tzv. *pipelined multilevel architecture* – PMA), koje koriste metod skeniranja piksela u parovima i koje omogućavaju procesiranje na visokoj radnoj frekvenciji, koje imaju malo kašnjenje i nizak nivo potrošnje u poređenju sa postojećim 2-D DWT arhitekturama. U istom radu je predložena i hardverski efikasna rekurzivna 2-D DWT arhitektura sa više nivoa dekompozicije/kompozicije (tzv. *recursive multilevel architecture* – RMA). Hardverski veoma efikasna 2-D DWT arhitektura, zasnovana na

dizanju i tehnici paralelne obrade podataka i savijanja, je predstavljena u radovima [75,76].

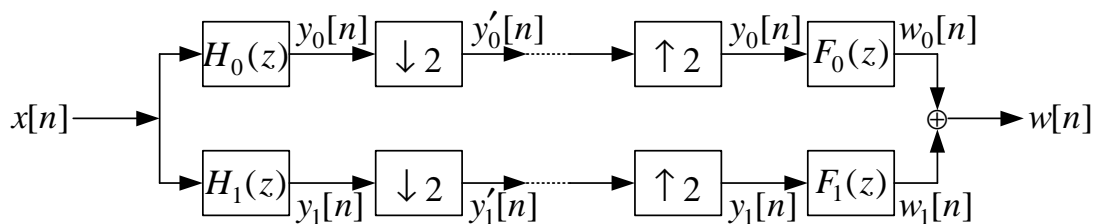
U pogledu utrošenih memorijskih resursa, 2-D DWT arhitekture [52,53], [62-64], [66], [68], [70-72], PMA [74] i [75,76] zahtevaju utrošak memorijskih resursa nižeg kapaciteta u odnosu na ostale arhitekture i ne zahtevaju korišćenje bilo kakvih eksternih memorija. Ukupno vreme procesiranja za 2-D DWT arhitekture [42], [50], [54], SIMD [58], [62,63], [65], [68,69], [72] i [74-76] je kraće nego za ostale pomenute arhitekture. Najmanje kašnjenje ima sistolno-paralelna arhitektura [59] i arhitekture [69,70] i [73,74], dok je kašnjenje za ostale arhitekture nekoliko puta veće, ali i dalje uporedivo sa pomenutim. U pogledu efikasnosti hardverskog iskorišćenja, najbolje performanse (efikasnost hardverskog iskorišćenja približno jednaku 1) imaju 2-D DWT arhitekture [42], [50], [65], [67], [69], FMA [74] i [75,76], dok preostale pomenute arhitekture imaju efikasnost hardverskog iskorišćenja manju od 1.

Neka od inicijalnih istraživanja u vezi sa hardverskom realizacijom inverznog podopsežnog transformatora, opisanog u ovoj disertaciji, su prezentovana u radovima [77-84].

4.2. Hardverska realizacija jednodimenzionalnog inverznog DWT 5/3 filtra

Blok dijagram tipičnog sistema za direktnu i inverznu jednodimenzionalnu (1-D) DWT, prema postojećem stanju tehnike, je prikazan na slici 4.1. Na strani direktne 1-D DWT, u gornjoj grani dijagrama sa slike 4.1, ulazni signal $x[n]$ se filtrira filtrom propusnikom niskih učestanosti sa funkcijom prenosa $H_0(z)$, nakon čega se vrši decimacija rezultujućih odbiraka sa faktorom dva. U donjoj grani dijagrama sa slike 4.1, takođe na strani direktne 1-D DWT, ulazni signal $x[n]$ se filtrira filtrom propusnikom visokih učestanosti sa funkcijom prenosa $H_1(z)$, nakon čega se vrši decimacija rezultujućih odbiraka sa faktorom dva. Na taj način se formiraju odbirci signala $y'_0[n]$ koji pripadaju niskopropusnom podopsegu i odbirci signala $y'_1[n]$ koji pripadaju

visokopropusnom podopsegu, koji se potom (uglavnom nakon kompresije) šalju kroz transmisioni kanal. Nakon prolaska kroz transmisioni kanal, na strani inverzne 1-D DWT, se u gornjoj grani dijagrama sa slike 4.1 (uglavnom nakon dekompresije) odbirci signala $y'_0[n]$ interpoliraju sa faktorom dva i filtriraju filtrom propusnikom niskih učestanosti sa funkcijom prenosa $F_0(z)$, čime se kreiraju rezultujući odbirci signala $w_0[n]$. Nakon prolaska kroz transmisioni kanal, takođe na strani inverzne 1-D DWT, se u donjoj grani dijagrama sa slike 4.1 (uglavnom nakon dekompresije) odbirci signala $y'_1[n]$ interpoliraju sa faktorom dva i filtriraju filtrom propusnikom visokih učestanosti sa funkcijom prenosa $F_1(z)$, čime se kreiraju rezultujući odbirci signala $w_1[n]$. Na osnovu odbiraka signala $w_0[n]$ i $w_1[n]$ se potom kreira rekonstruisani signal $w[n]$.



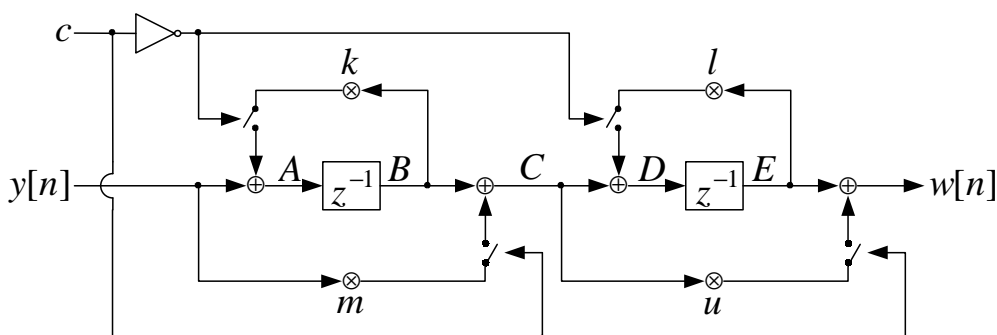
Slika 4.1. Blok dijagram tipičnog sistema za direktnu i inverznu 1-D DWT prema postojećem stanju tehnike

Svaki drugi odbirak i u gornjoj i u donjoj grani dijagrama sa slike 4.1 se na strani inverzne 1-D DWT dodaje tokom procesa interpolacije, povećavajući na taj način utrošak memorijskih resursa i resursa za procesiranje potrebnih za realizaciju inverzne 1-D DWT. Ovakav pristup ima za posledicu da se i odbirci koji su dodati procesom interpolacije filtriraju filterima $F_0(z)$ i $F_1(z)$ iako to nije neophodno. Time se bespotrebno troše memorijski i logički resursi filtera $F_0(z)$ i $F_1(z)$, koji se koriste za filtriranje pomenutih odbiraka.

Pristup [85-87] opisan u ovoj disertaciji se zasniva na ideji da se logički i memorijski resursi 1-D inverznih DWT 5/3 filtera, koji se bespotrebno troše u tipičnim realizacijama prema postojećem stanju tehnike, iskoriste isključivo za filtriranje odbiraka podataka koji nisu dodati procesom interpolacije. U tom novom konceptu

parni vremenski slotovi se koriste za filtriranje odbiraka signala koji pripadaju niskopropusnom podopsegu, dok se neparni vremenski slotovi koriste za filtriranje odbiraka signala koji pripadaju visokopropusnom podopsegu. Na taj način se u realizaciji 1-D inverznog DWT 5/3 filtra opisanoj u ovoj disertaciji koriste logički i memorijski resursi koji se bespotrebno troše u tipičnim realizacijama 1-D inverznog DWT 5/3 filtra prema postojećem stanju tehnike.

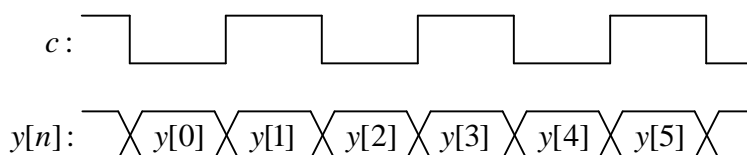
Dodatne uštede memorijskih resursa u realizaciji 1-D inverznog DWT 5/3 filtra opisanoj u ovoj disertaciji, u poređenju sa tipičnim realizacijama prema postojećem stanju tehnike, postižu se korišćenjem istih registara za proces filtriranja odbiraka signala koji pripadaju niskopropusnom podopsegu i za proces filtriranja odbiraka signala koji pripadaju visokopropusnom podopsegu, što je izvodljivo pošto se niskopropusne i visokopropusne komponente signala filtriraju u različitim vremenskim slotovima. Međutim, s obzirom da se funkcija prenosa filtra propusnika niskih učestanosti razlikuje od funkcije prenosa filtra propusnika visokih učestanosti, pristup opisan u ovoj disertaciji zahteva korišćenje vremenski nestacionarne topologije za realizaciju 1-D inverznog DWT 5/3 filtra. Jedna konfiguracija filtra se koristi za filtriranje niskopropusnih komponenti signala (tokom parnih vremenskih slotova), dok se druga konfiguracija filtra koristi za filtriranje visokopropusnih komponenti signala (tokom neparnih vremenskih slotova). Promena konfiguracije vremenski nestacionarne topologije filtra opisane u ovoj disertaciji se postiže korišćenjem prekidača, pri čemu se upotreba istih registara za filtriranje niskopropusnih i visokopropusnih komponenti signala omogućava korišćenjem tzv. *feed-forward* i *feedback* putanja.



Slika 4.2. Šematski blok dijagram realizacije 1-D inverznog DWT 5/3 filtra opisanog u ovoj disertaciji

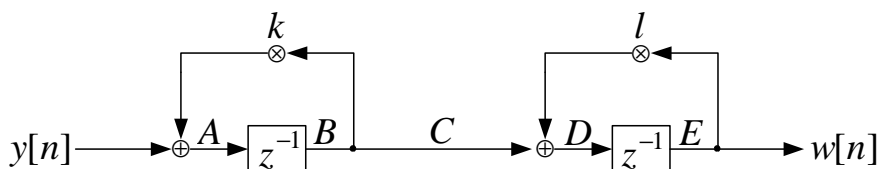
Šematski blok dijagram realizacije 1-D inverznog DWT 5/3 filtra iz ove disertacije je prikazan na slici 4.2.

Kontrolni signal c se koristi za upravljanje prekidačima, koji omogućavaju realizaciju vremenski nestacionarne topologije filtra. Vremenski dijagram kontrolnog signala c je prikazan na slici 4.3.



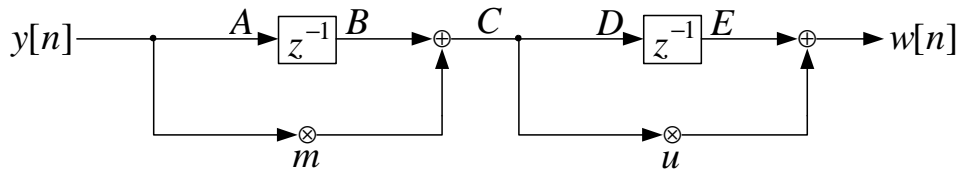
Slika 4.3. Vremenski dijagram kontrolnog signala c u realizaciji 1-D inverznog DWT 5/3 filtra opisanoj u ovoj disertaciji

Kada god je kontrolni signal c na niskom logičkom nivou ($c=0$), tj. za svaki ulazni odbirak $y[n]$ sa parnim indeksom $n=2p$, dva gornja prekidača su zatvorena, a dva donja prekidača su otvorena, što daje konfiguraciju filtra prikazanu na slici 4.4.



Slika 4.4. Konfiguracija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji za $c=0$

Kada god je kontrolni signal c na visokom logičkom nivou ($c=1$), tj. za svaki ulazni odbirak $y[n]$ sa neparnim indeksom $n=2p+1$, dva gornja prekidača su otvorena, a dva donja prekidača su zatvorena, što daje konfiguraciju filtra prikazanu na slici 4.5.



Slika 4.5. Konfiguracija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji za $c=1$

Set jednačina (4.5)-(4.11) opisuje signale unutar realizacije 1-D inverznog DWT 5/3 filtra iz ove disertacije, u vremenskim trenucima od $n=0$ do $n=5$.

U trenutku $n=0$, 1-D inverzni DWT 5/3 filter iz ove disertacije ima konfiguraciju prikazanu na slici 4.4, tako da važi jednakost:

$$A = y[n]. \quad (4.5)$$

U trenutku $n=1$, 1-D inverzni DWT 5/3 filter iz ove disertacije ima konfiguraciju prikazanu na slici 4.5, tako da važe sledeće jednakosti:

$$\begin{aligned} B &= y[n-1], \\ A &= y[n], \\ C &= m \cdot y[n] + y[n-1] = D. \end{aligned} \quad (4.6)$$

Za $n=2$, 1-D inverzni DWT 5/3 filter iz ove disertacije ponovo ima konfiguraciju prikazanu na slici 4.4, što daje sledeće jednačine:

$$\begin{aligned} B &= y[n-1] = C, \\ A &= y[n] + k \cdot y[n-1], \\ E &= m \cdot y[n-1] + y[n-2], \\ D &= y[n-1] + m \cdot l \cdot y[n-1] + l \cdot y[n-2]. \end{aligned} \quad (4.7)$$

Za $n=3$, 1-D inverzni DWT 5/3 filter iz ove disertacije ponovo ima konfiguraciju prikazanu na slici 4.5, iz čega se mogu izvesti sledeće jednačine:

$$\begin{aligned}
B &= y[n-1] + k \cdot y[n-2], \\
C &= m \cdot y[n] + y[n-1] + k \cdot y[n-2] = D, \\
E &= y[n-2] + m \cdot l \cdot y[n-2] + l \cdot y[n-3], \\
A &= y[n], \\
w[n] &= y[n-2] + m \cdot l \cdot y[n-2] + l \cdot y[n-3], \\
&\quad + m \cdot u \cdot y[n] + u \cdot y[n-1] + k \cdot u \cdot y[n-2].
\end{aligned} \tag{4.8}$$

Za $n=4$, 1-D inverzni DWT 5/3 filter iz ove disertacije ponovo ima konfiguraciju prikazanu na slici 4.4, iz čega sledi:

$$\begin{aligned}
B &= y[n-1] = C, \\
A &= y[n] + k \cdot y[n-1], \\
E &= m \cdot y[n-1] + y[n-2] + k \cdot y[n-3] = w[n], \\
D &= y[n-1] + l \cdot m \cdot y[n-1] + l \cdot y[n-2] + k \cdot l \cdot y[n-3].
\end{aligned} \tag{4.9}$$

Za $n=5$, 1-D inverzni DWT 5/3 filter iz ove disertacije ponovo ima konfiguraciju prikazanu na slici 4.5, što daje:

$$\begin{aligned}
B &= y[n-1] + k \cdot y[n-2], \\
A &= y[n], \\
C &= m \cdot y[n] + y[n-1] + k \cdot y[n-2], \\
E &= y[n-2] + l \cdot m \cdot y[n-2] + l \cdot y[n-3] + k \cdot l \cdot y[n-4], \\
w[n] &= y[n-2] + l \cdot m \cdot y[n-2] + l \cdot y[n-3] + k \cdot l \cdot y[n-4] \\
&\quad + m \cdot u \cdot y[n] + u \cdot y[n-1] + k \cdot u \cdot y[n-2].
\end{aligned} \tag{4.10}$$

Na osnovu seta jednačina (4.10), $w[n]$ se može izraziti na sledeći način:

$$\begin{aligned}
w[n] &= m \cdot u \cdot y[n] + u \cdot y[n-1] + (1 + l \cdot m + k \cdot u) \cdot y[n-2] \\
&\quad + l \cdot y[n-3] + k \cdot l \cdot y[n-4].
\end{aligned} \tag{4.11}$$

Na dalje, opisana situacija se periodično ponavlja. Za svaki neparni indeks n , $w[n]$ zadovoljava jednačinu (4.11), dok za svaki parni indeks n , $w[n]$ može da se izrazi na sledeći način:

$$w[n] = m \cdot y[n-1] + y[n-2] + k \cdot y[n-3]. \quad (4.12)$$

Da bi se odredile vrednosti parametara k , l , m i u , potrebno je iskoristiti činjenicu da je $w[n]$ zapravo zakašnjena replika sekvence $x[n]$, s obzirom da 1-D inverzni DWT 5/3 filter treba da obezbedi savršenu rekonstruktiju odbiraka signala, prethodno filtriranih pomoću 1-D direktnog DWT 5/3 filtra, tj.

$$w[n] = x[n - \delta]. \quad (4.13)$$

Parametar δ predstavlja kašnjenje koje proizvodi kaskadna veza 1-D direktnog DWT 5/3 filtra i 1-D inverznog DWT 5/3 filtra.

Za svaki parni indeks n , $y[n]$ zadovoljava diferencnu jednačinu (4.1), dok za svaki neparni indeks n , $y[n]$ zadovoljava diferencnu jednačinu (4.2). Ako se zamene $y[n-1]$ i $y[n-3]$ iz jednačine (4.12) sa odgovarajućom diferencnom jednačinom (4.2), i ako se zameni $y[n-2]$ iz jednačine (4.12) sa odgovarajućom diferencnom jednačinom (4.1), kao rezultat se dobija:

$$\begin{aligned} w[n] = & -\frac{1}{2}m \cdot x[n-2] + m \cdot x[n-3] - \frac{1}{2}m \cdot x[n-4] \\ & - \frac{1}{8}x[n-2] + \frac{1}{4}x[n-3] + \frac{3}{4}x[n-4] + \frac{1}{4}x[n-5] \\ & - \frac{1}{8}x[n-6] - \frac{1}{2}k \cdot x[n-4] + k \cdot x[n-5] - \frac{1}{2}k \cdot x[n-6]. \end{aligned} \quad (4.14)$$

Na osnovu prethodne jednačine, $w[n]$ se može izraziti na sledeći način:

$$\begin{aligned} w[n] = & \left(-\frac{1}{2}m - \frac{1}{8}\right) \cdot x[n-2] + \left(m + \frac{1}{4}\right) \cdot x[n-3] \\ & + \left(-\frac{1}{2}m + \frac{3}{4} - \frac{1}{2}k\right) \cdot x[n-4] + \left(\frac{1}{4} + k\right) \cdot x[n-5] + \left(-\frac{1}{8} - \frac{1}{2}k\right) \cdot x[n-6]. \end{aligned} \quad (4.15)$$

Iz prethodne jednačine se može zaključiti da odbirci signala $x[n-2]$, $x[n-3]$, $x[n-5]$ i $x[n-6]$ isčezavaju iz jednačine ako se izabere da je:

$$m = k = -\frac{1}{4}. \quad (4.16)$$

U tom slučaju jednačina (4.15) poprima sledeći oblik:

$$w[n] = x[n-4], \quad (4.17)$$

što znači da je ostvarena savršena rekonstrukcija signala za parne indekse n .

Ako se vrednosti za m i k iz jednačine (4.16) zamene u jednačinu (4.11) koja određuje vrednosti $w[n]$ za neparne indekse n , dobija se:

$$\begin{aligned} w[n] = & -\frac{1}{4}u \cdot y[n] + u \cdot y[n-1] \\ & + \left(1 - \frac{1}{4}l - \frac{1}{4}u\right) \cdot y[n-2] + l \cdot y[n-3] - \frac{1}{4}l \cdot y[n-4]. \end{aligned} \quad (4.18)$$

Ako se zamene $y[n]$, $y[n-2]$ i $y[n-4]$ iz jednačine (4.18) sa odgovarajućom diferencnom jednačinom (4.2), i ako se zamene $y[n-1]$ i $y[n-3]$ iz jednačine (4.18) sa odgovarajućom diferencnom jednačinom (4.1), kao rezultat se dobija:

$$\begin{aligned} w[n] = & \frac{1}{8}u \cdot x[n-1] - \frac{1}{4}u \cdot x[n-2] + \frac{1}{8}u \cdot x[n-3] \\ & - \frac{1}{8}u \cdot x[n-1] + \frac{1}{4}u \cdot x[n-2] + \frac{3}{4}u \cdot x[n-3] + \frac{1}{4}u \cdot x[n-4] \\ & - \frac{1}{8}u \cdot x[n-5] + \left(-\frac{1}{2} + \frac{1}{8}l + \frac{1}{8}u\right) \cdot x[n-3] + \left(1 - \frac{1}{4}l - \frac{1}{4}u\right) \cdot x[n-4] \\ & + \left(-\frac{1}{2} + \frac{1}{8}l + \frac{1}{8}u\right) \cdot x[n-5] - \frac{1}{8}l \cdot x[n-3] + \frac{1}{4}l \cdot x[n-4] + \frac{3}{4}l \cdot x[n-5] \\ & + \frac{1}{4}l \cdot x[n-6] - \frac{1}{8}l \cdot x[n-7] + \frac{1}{8}l \cdot x[n-5] - \frac{1}{4}l \cdot x[n-6] + \frac{1}{8}l \cdot x[n-7]. \end{aligned} \quad (4.19)$$

Na osnovu prethodne jednačine $w[n]$ se konačno može izraziti kao:

$$w[n] = \left(u - \frac{1}{2}\right) \cdot x[n-3] + x[n-4] + \left(l - \frac{1}{2}\right) \cdot x[n-5]. \quad (4.20)$$

Iz jednačine (4.20) se može zaključiti da odbirci signala $x[n-3]$ i $x[n-5]$ iščezavaju iz jednačine ako se izabere da je:

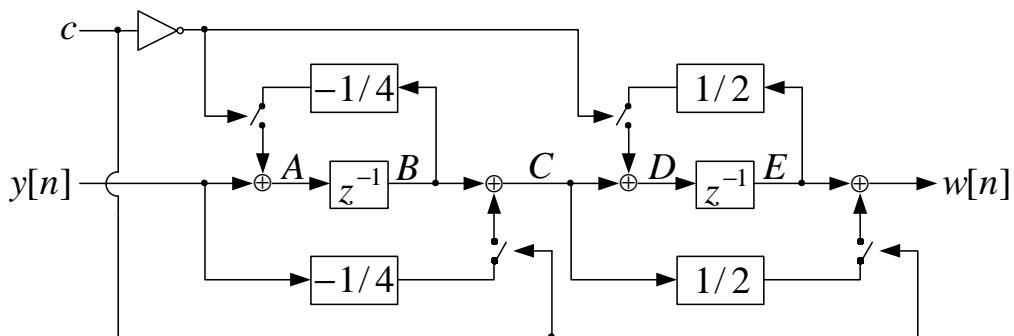
$$u = l = \frac{1}{2}. \quad (4.21)$$

U tom slučaju jednačina (4.20) poprima sledeći oblik:

$$w[n] = x[n-4], \quad (4.22)$$

što znači da je ostvarena savršena rekonstrukcija signala i za neparne indekse n .

Stoga, realizacija 1-D inverznog DWT 5/3 filtra iz ove disertacije, prikazana na slici 4.6, radi kao filter propusnik niskih učestanosti za odbirke signala $y[n]$ sa parnim indeksima n , a kao filter propusnik visokih učestanosti za odbirke signala $y[n]$ sa neparnim indeksima n .



Slika 4.6. Realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji

Zamena dobijenih vrednosti za m i k iz jednačine (4.16) u jednačinu (4.12) obezbeđuje da se ulazni odbirci $y[n]$ niskopropusno filtriraju u vremenskim slotovima sa parnim indeksima $n=2p$ bez potrebe da se vrši interpolacija sa faktorom dva, pri čemu se generišu izlazni odbirci $w_0[n]$, koji zapravo predstavljaju odbirke izlaznog signala $w[n]$ sa parnim indeksima $n=2p$ (4.23).

$$w_0[n] = -\frac{1}{4}y[n-1] + y[n-2] - \frac{1}{4}y[n-3]. \quad (4.23)$$

Zamena dobijenih vrednosti za k , l , m i u iz jednačina (4.16) i (4.21) u jednačinu (4.11) obezbeđuje da se ulazni odbirci $y[n]$ visokopropusno filtriraju u vremenskim slotovima sa neparnim indeksima $n=2p+1$ bez potrebe da se vrši interpolacija sa faktorom dva, pri čemu se generišu izlazni odbirci $w_1[n]$, koji zapravo predstavljaju odbirke izlaznog signala $w[n]$ sa neparnim indeksima $n=2p+1$ (4.24).

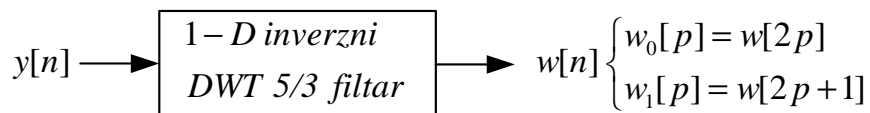
$$w_1[n] = -\frac{1}{8}y[n] + \frac{1}{2}y[n-1] + \frac{3}{4}y[n-2] + \frac{1}{2}y[n-3] - \frac{1}{8}y[n-4]. \quad (4.24)$$

Diferencna jednačina (4.23) se razlikuje od diferencne jednačine (4.3), što je i očekivano s obzirom da jednačina (4.3) sadrži samo odbirke signala $y_0[n]$ generisane direktnim niskopropusnim filtriranjem odbiraka ulaznog signala $x[n]$, dok jednačina (4.23) sadrži odbirke signala $y[n]$ koji zapravo predstavljaju učešljane odbirke signala generisane direktnim niskopropusnim filtriranjem i direktnim visokopropusnim filtriranjem odbiraka ulaznog signala $x[n]$.

Slično tome, diferencna jednačina (4.24) se razlikuje od diferencne jednačine (4.4), što je i očekivano s obzirom da jednačina (4.4) sadrži samo odbirke signala $y_1[n]$ generisane direktnim visokopropusnim filtriranjem odbiraka ulaznog signala $x[n]$, dok jednačina (4.24) sadrži odbirke signala $y[n]$ koji zapravo predstavljaju učešljane odbirke signala generisane direktnim niskopropusnim filtriranjem i direktnim visokopropusnim filtriranjem odbiraka ulaznog signala $x[n]$.

Međutim, obe diferencne jednačine (4.23) i (4.24) se u potpunosti poklapaju sa odgovarajućim jednačinama za odbirke rekonstruisanog signala za 1-D inverzni DWT 5/3 filter zasnovan na dizanju, pošto taj filter generiše odbirke izlaznih signala $w_0[n]$ i $w_1[n]$ na isti način kao i filter opisan u ovoj disertaciji, tj. na bazi odbiraka signala $y[n]$ koji predstavljaju učešljane odbirke signala generisane direktnim niskopropusnim filtriranjem i direktnim visokopropusnim filtriranjem odbiraka ulaznog signala $x[n]$.

Dakle, može se zaključiti da hardverska realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji obezbeđuje savršenu rekonstrukciju signala na izlazu i za odbirke sa parnim indeksima $n = 2p$ i za odbirke sa neparnim indeksima $n = 2p + 1$, koji se pri tome učešljavaju bez ikakve dodatne operacije sabiranja (slika 4.7).

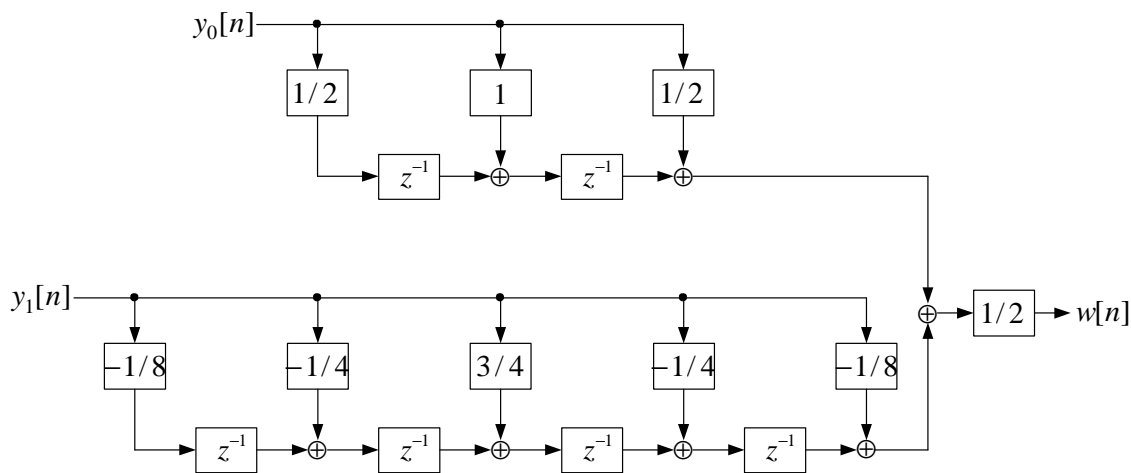


Slika 4.7. Blok dijagram realizacije 1-D inverznog DWT 5/3 filtra opisane u ovoj disertaciji

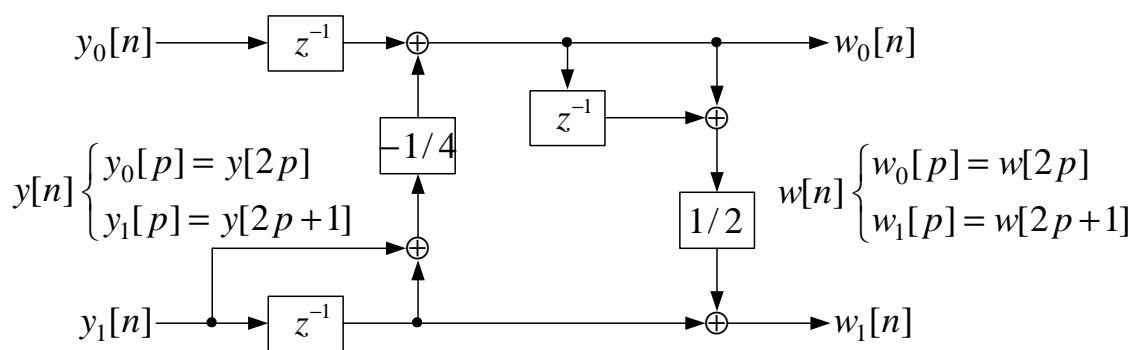
4.2.1. Poređenja performansi različitih hardverskih realizacija jednodimenzionalnog inverznog DWT 5/3 filtra

Hardverska realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji (slika 4.6) je upoređena u pogledu hardverske kompleksnosti sa najpoznatijim i najboljim hardverskim realizacijama 1-D inverznog DWT 5/3 filtra prema postojećem stanju tehnike i to sa realizacijom zasnovanom na konvoluciji [41-46] (prikazanom na slici 4.8) i najefikasnijom realizacijom [47-50], [58,59] (prikazanom na slici 4.9) među realizacijama zasnovanim na dizanju. Iako je realizacija [50] nešto kompleksnija nego što je to prikazano na slici 4.9, ona se može redukovati na formu prikazanu na slici 4.9 uklanjanjem *pipeline* registara. U radu [57] je pokazano da, kao cena plaćena za veću fleksibilnost dizajna koja uključuje i podršku za širok spektar različitih tipova filtera,

realizacije zasnovane na dizanju [51-55] imaju veću hardversku kompleksnost nego realizacije prezentovane u [47-50] i [58,59]. Iz tog razloga, poređenje sa realizacijama [51-55] nije prikazano u ovoj disertaciji.



Slika 4.8. Realizacija 1-D inverznog DWT 5/3 filtra zasnovana na konvoluciji prema postojećem stanju tehnike



Slika 4.9. Realizacija 1-D inverznog DWT 5/3 filtra zasnovana na dizanju prema postojećem stanju tehnike

U tabeli 4.1 je prikazan pregled korišćenih hardverskih komponenti za pomenute realizacije 1-D inverznog DWT 5/3 filtra. Tabela 4.1 ne sadrži informacije o broju korišćenih množača, pošto množači mogu u potpunosti da budu izbegnuti u svim navedenim realizacijama zbog specifičnih vrednosti koeficijenata sa kojima se vrši

množenje. Naime, svi koeficijenti sa kojima se vrši množenje se mogu predstaviti preko stepena broja dva (za realizaciju zasnovanu na dizanju prema postojećem stanju tehnike i realizaciju opisanu u ovoj disertaciji) ili preko sume brojeva koji se mogu predstaviti preko stepena broja dva (za realizaciju zasnovanu na konvoluciji prema postojećem stanju tehnike). Iz tog razloga se u svim pomenutim realizacijama umesto množača koriste bitski pomerene hardverske konekcije između ulaznih i izlaznih bitskih linija.

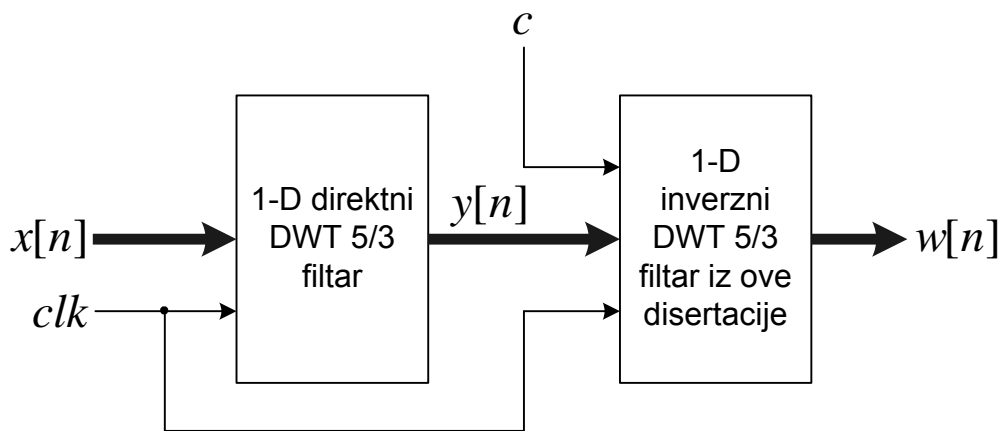
Tabela 4.1. Korišćene hardverske komponente za različite realizacije 1-D inverznog DWT 5/3 filtra

Komponenta / Realizacija filtra	Konvolucija [41-46]	Dizanje [47-50], [58,59]	Realizacija iz ove disertacije
Sabirači	7	4	4
Elementi za kašnjenje	6	3	2
Dodatna logika za učešljavanje izlaznih podataka	Nije potrebna	Potrebna je	Nije potrebna

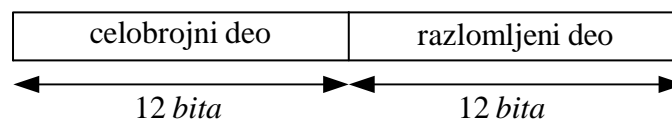
Može se uočiti da je realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji najjednostavnija i da zahteva korišćenje najmanjeg broja hardverskih komponenti. Pored toga, dok realizacija 1-D inverznog DWT 5/3 filtra zasnovana na dizanju prema postojećem stanju tehnike zahteva korišćenje dodatne logike za učešljavanje izlaznih odbiraka signala $w_0[n]$ i $w_1[n]$ u slučajevima kada izlazni odbirci sa parnim i neparnim indeksima treba da budu generisani serijski u naizmeničnim ciklusima takta, realizacija opisana u ovoj disertaciji na izlazu generiše učešljane odbirke bez korišćenja ikakve dodatne logike. Stoga realizacije iz ove disertacije predstavlja najoptimalnije rešenje u pogledu jednostavnosti i uštede korišćenih resursa.

4.2.2. Poređenja rezultata sinteze različitih hardverskih realizacija jednodimenzionalnog inverznog DWT 5/3 filtra

Funkcionalna verifikacija hardverske realizacije 1-D inverznog DWT 5/3 filtra opisane u ovoj disertaciji je izvršena pomoću *FPGA* implementacije kaskadne veze 1-D direktnog DWT 5/3 filtra i pomenute realizacije 1-D inverznog DWT 5/3 filtra iz ove disertacije (slika 4.10).



Slika 4.10. Kaskadna veza 1-D direktnog DWT 5/3 filtra i realizacije 1-D inverznog DWT 5/3 filtra opisane u ovoj disertaciji



Slika 4.11. Format podataka korišćen za funkcionalnu verifikaciju realizacije 1-D inverznog DWT 5/3 filtra opisane u ovoj disertaciji

Ova kaskadna struktura je korišćenjem softverskog okruženja *Altera Quartus II* implementirana za 24-bitne podatke u formatu sa fiksnom tačkom (sa 12 bita namenjenih za predstavljanje celobrojnog dela podatka i 12 bita namenjenih za predstavljanje razlomljenog dela podatka, kao što je to prikazano na slici 4.11), jer ovaj

format podataka obezbeđuje ispravno predstavljanje generisanih odbiraka signala za bar 4 nivoa dekompozicije kod DWT.

Za potrebe funkcionalne verifikacije korišćena je proizvoljna sekvenca ulaznih podataka $x[n]$. Simulacija kaskadne strukture je izvršena pomoću *Simulator Tool* iz softverskog paketa *Altera Quartus II* i simulacioni rezultati su potvrdili funkcionalnu ispravnost realizacije 1-D inverznog DWT 5/3 filtra opisane u ovoj disertaciji, pošto su rezultujući podaci na izlazu $w[n]$ bili zakašnjenja replika sekvence ulaznih podataka $x[n]$ (sa kašnjenjem od 4 periode takta). Imajući u vidu da se efikasnost hardverskog iskorišćenja može definisati kao odnos broja perioda takta u kojima se vrši izračunavanje vrednosti izlaznih podataka i ukupnog broja perioda takta koliko traje procesiranje podataka, može se zaključiti da realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji ima efikasnost hardverskog iskorišćenja 100%.

Tabela 4.2. Performanse različitih realizacija 1-D inverznog DWT 5/3 filtra implementiranih u *Altera FPGA* čipu EP4CE115F29C7

1-D inverzni DWT 5/3 filter na 85°C	Konvolucija [41-46]	Dizanje [47-50,58,59]	Dizanje+ [47-50,58,59]	Realizacija iz ove disertacije
Broj iskorišćenih logičkih elemenata	234	120	144	120
Broj iskorišćenih registara	139	72	72	48
Maksimalna radna frekvencija [MHz]	197,7	128	197,3	212
Ukupna disipacija [mW] na 80MHz	132,4	134,4	177,7	130,9

Rezultati sinteze različitih hardverskih realizacija 1-D inverznog DWT 5/3 filtra dobijeni korišćenjem softverskog paketa *Altera Quartus II* 10.0 za *FPGA* čip EP4CE115F29C7 su prezentovani u tabeli 4.2. Druga kolona te tabele sadrži podatke koji se odnose na realizaciju 1-D inverznog DWT 5/3 filtra zasnovanu na konvoluciji prema postojećem stanju tehnike [41-46], treća kolona sadrži podatke koji se odnose na najefikasniju među realizacijama 1-D inverznog DWT 5/3 filtra zasnovanima na dizanju prema postojećem stanju tehnike [47-50] i [58,59] bez bilo kakve dodatne logike za učešljavanje odbiraka izlaznog signala, dok četvrta kolona sadrži podatke koji se odnose na najefikasniju među realizacijama 1-D inverznog DWT 5/3 filtra zasnovanima na

dizanju prema postojećem stanju tehnike [47-50] i [58,59] sa dodatnom logikom za učešljavanje odbiraka izlaznog signala. Konačno, u petoj koloni su predstavljeni rezultati sinteze za realizaciju 1-D inverznog DWT 5/3 filtra opisanu u ovoj disertaciji.

Rezultati sinteze jasno pokazuju da je realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji, u pogledu ukupnog broja korišćenih registara, 65% jednostavnija nego realizacija 1-D inverznog DWT 5/3 filtra zasnovana na konvoluciji i 33% jednostavnija od realizacija 1-D inverznog DWT 5/3 filtra zasnovanih na dizanju (sa ili bez dodatne logike za učešljavanje odbiraka izlaznog signala). U pogledu ukupnog broja korišćenih logičkih elemenata, realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji je 49% jednostavnija od realizacije 1-D inverznog DWT 5/3 filtra zasnovane na konvoluciji, dok sa druge strane ima istu kompleksnost kao i realizacija 1-D inverznog DWT 5/3 filtra zasnovana na dizanju bez dodatne logike za učešljavanje odbiraka izlaznog signala. U slučajevima kada je dodatna logika za učešljavanje odbiraka izlaznog signala neophodna, realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji je 17% jednostavnija od odgovarajuće realizacije zasnovane na dizanju, u pogledu ukupnog broja korišćenih logičkih elemenata.

Maksimalna radna frekvencija realizacije 1-D inverznog DWT 5/3 filtra opisane u ovoj disertaciji je 7% veća u poređenju sa realizacijom zasnovanom na konvoluciji i realizacijom zasnovanom na dizanju sa dodatnom logikom za učešljavanje odbiraka izlaznog signala, a čak 40% veća u poređenju sa realizacijom zasnovanom na dizanju bez dodatne logike za učešljavanje odbiraka izlaznog signala. Maksimalne radne frekvencije za ove implementacije su određene pri najgorim operativnim uslovima – pri najvećoj radnoj temperaturi (85°C) i najsporijem silicijumu u čipu. Tabela 4.2 takođe pokazuje da realizacija 1-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji ima najmanju ukupnu disipaciju u poređenju sa ostale tri realizacije 1-D inverznog DWT 5/3 filtra.

4.3. Hardverska realizacija dvodimenzionalnog inverznog DWT 5/3 filtra

Struktura 2-D inverzne DWT 5/3 arhitekture sa $J = 7$ nivoa kompozicije čija će realizacija biti opisana u ovoj sekciji, koja koristi 1-D inverzni DWT 5/3 filtar opisan u sekciji 4.2 ove disertacije, prikazana je na slici 4.12. Arhitektura koja podržava sedam nivoa kompozicije je prikazana iz razloga što ovaj broj nivoa kompozicije omogućava odličan kvalitet kompresije i dekompresije za slike *high-definition* (HD) rezolucije (1920x1080 piksela). Ova struktura se može veoma jednostavno modifikovati sa ciljem da podrži i bilo koji drugi broj nivoa kompozicije.

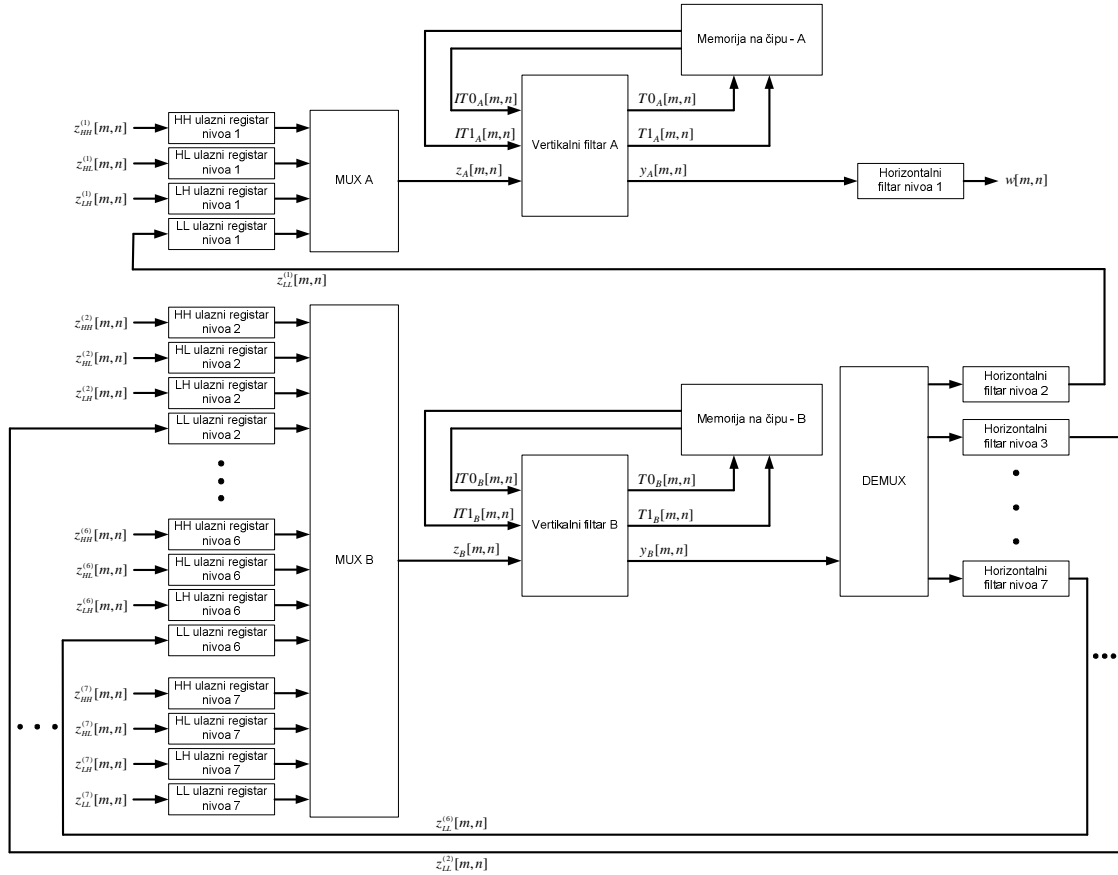
Ulazni podaci za 2-D inverznu DWT 5/3 arhitekturu opisanu u ovoj disertaciji su komponente dekomponovanog signala $z_{HH}^{(j)}[m,n]$, $z_{HL}^{(j)}[m,n]$ i $z_{LH}^{(j)}[m,n]$ iz nivoa j ($j=1,2,\dots,7$) kompozicije, kao i komponente dekomponovanog signala $z_{LL}^{(7)}[m,n]$ iz nivoa 7 kompozicije.

Komponente $z_{HH}^{(j)}[m,n]$, $z_{HL}^{(j)}[m,n]$, $z_{LH}^{(j)}[m,n]$ i $z_{LL}^{(j)}[m,n]$ koje pripadaju podopsezima HH, HL, LH i LL, respektivno, nivoa j ($j=2,3,\dots,7$) se prihvataju u odgovarajuće registre "Ulazni registar nivoa j ", a zatim se procesiraju multiplekserom na čijem izlazu se pojavljuju odbirci signala $z_B[m,n]$:

$$z_B[m,n] = \begin{cases} z_{LH}^{(j)}[m,k], & \text{za } m = 2l \text{ i } n = 2k \\ z_{LL}^{(j)}[m,k], & \text{za } m = 2l \text{ i } n = 2k + 1 \\ z_{HH}^{(j)}[m,k], & \text{za } m = 2l + 1 \text{ i } n = 2k \\ z_{HL}^{(j)}[m,k], & \text{za } m = 2l + 1 \text{ i } n = 2k + 1 \end{cases} \quad (4.25)$$

U ovoj notaciji podopseg LL predstavlja rezultat direktnog niskofrekventnog filtriranja duž vrsta i direktnog niskofrekventnog filtriranja duž kolona. Podopseg HL predstavlja rezultat direktnog niskofrekventnog filtriranja duž vrsta i direktnog visokofrekventnog filtriranja duž kolona. Podopseg LH predstavlja rezultat direktnog visokofrekventnog filtriranja duž vrsta i direktnog niskofrekventnog filtriranja duž

kolona. Konačno, podopseg HH predstavlja rezultat direktnog visokofrekventnog filtriranja duž vrsta i direktnog visokofrekventnog filtriranja duž kolona.



Slika 4.12. Realizacija 2-D inverznog DWT 5/3 filtra opisana u ovoj disertaciji

Odbirci signala $z_B[m,n]$ se dalje vertikalno filtriraju pomoću “Vertikalnog filtra B”, koji na svom izlazu generiše odbirke signala $y_B[m,n]$:

$$y_B[m,n] = \begin{cases} y_H^{(j)}[m,k], & \text{za } n = 2k \\ y_L^{(j)}[m,k], & \text{za } n = 2k + 1 \end{cases}, \quad (4.26)$$

gde $y_H^{(j)}[m,k]$ predstavlja visokofrekventne komponente signala iz nivoa j ($j=2,3,\dots,7$) koje potom treba da se horizontalno filtriraju, a $y_L^{(j)}[m,k]$ predstavlja niskofrekventne komponente signala iz nivoa j ($j=2,3,\dots,7$) koje treba da se horizontalno filtriraju.

Odbirci signala $y_B[m,n]$ se zatim propuštaju kroz demultiplekser i horizontalno filtriraju pomoću “Horizontalnog filtra nivoa j ” ($j=2,3,\dots,7$), koji na svom izlazu generiše komponente dekomponovanog signala $z_{LL}^{(j-1)}[m,n]$ ($j=2,3,\dots,7$) koje će kasnije biti iskorišćene za inverzno filtriranje na nivou $j-1$. Svi horizontalni filtri su implementirani u vidu realizacija 1-D inverznog DWT 5/3 filtra iz sekcije 4.2.

Komponente $z_{HH}^{(1)}[m,n]$, $z_{HL}^{(1)}[m,n]$, $z_{LH}^{(1)}[m,n]$ i $z_{LL}^{(1)}[m,n]$ koje pripadaju podopsezima HH, HL, LH i LL, respektivno, nivoa 1 se prihvataju u odgovarajuće registre “Ulazni registar nivoa 1”, a zatim se procesiraju multiplekserom na čijem izlazu se pojavljuju odbirci signala $z_A[m,n]$:

$$z_A[m,n] = \begin{cases} z_{LH}^{(1)}[m,k], & \text{za } m=2l \text{ i } n=2k \\ z_{LL}^{(1)}[m,k], & \text{za } m=2l \text{ i } n=2k+1 \\ z_{HH}^{(1)}[m,k], & \text{za } m=2l+1 \text{ i } n=2k \\ z_{HL}^{(1)}[m,k], & \text{za } m=2l+1 \text{ i } n=2k+1 \end{cases} \quad (4.27)$$

Odbirci signala $z_A[m,n]$ se dalje vertikalno filtriraju pomoću “Vertikalnog filtra A”, koji na svom izlazu generiše odbirke signala $y_A[m,n]$:

$$y_A[m,n] = \begin{cases} y_H^{(1)}[m,k], & \text{za } n=2k \\ y_L^{(1)}[m,k], & \text{za } n=2k+1 \end{cases} \quad (4.28)$$

gde $y_H^{(1)}[m,k]$ predstavlja visokofrekventne komponente signala iz nivoa 1 koje potom treba da se horizontalno filtriraju, a $y_L^{(1)}[m,k]$ predstavlja niskofrekventne komponente signala iz nivoa 1 koje treba da se horizontalno filtriraju. Odbirci signala $y_A[m,n]$ se zatim horizontalno filtriraju pomoću “Horizontalnog filtra nivoa 1”, koji na svom izlazu generiše piksele rekonstruisane slike $w[m,n]$. “Horizontalni filter nivoa 1” je implementiran u vidu realizacije 1-D inverznog DWT 5/3 filtra iz sekcije 4.2.

Dinamika 2-D inverznog filtriranja na početku parnih linija (brojeći od 0) je predstavljena na vremenskom dijagramu prikazanom na slici 4.13. Ovaj vremenski dijagram prikazuje linije iz prva tri nivoa kompozicije u slučaju kada su linije na

svakom od tih nivoa parne linije (pošto samo parne linije sadrže komponente dekomponovanog signala iz LL podopsega, a upravo ove komponente su neophodne za dalje inverzno filtriranje na narednom nivou kompozicije). Prikazana su samo tri nivoa kompozicije iz razloga da bi se smanjila kompleksnost dijagrama, ali obrazac prikazan na slici 4.13 se jednostavno može proširiti i na proizvoljan broj nivoa kompozicije.

takt	VF ulaz nivo 3	HF ulaz nivo 3	VF ulaz nivo 2	HF ulaz nivo 2	VF ulaz nivo 1	HF ulaz nivo 1	Izlazni pikseli
0	$z_{LH}^{(3)}[m^{(3)},0]$	$y_H^{(3)}[m^{(3)},0]$					
1	$z_{LL}^{(3)}[m^{(3)},0]$	$y_L^{(3)}[m^{(3)},0]$					
2	$z_{LH}^{(3)}[m^{(3)},1]$	$y_H^{(3)}[m^{(3)},1]$					
3	$z_{LL}^{(3)}[m^{(3)},1]$	$y_L^{(3)}[m^{(3)},1]$					
4	$z_{LH}^{(3)}[m^{(3)},2]$	$y_H^{(3)}[m^{(3)},2]$					
5			$z_{LH}^{(2)}[m^{(2)},0]$	$y_H^{(2)}[m^{(2)},0]$			
6			$z_{LL}^{(2)}[m^{(2)},0]$	$y_L^{(2)}[m^{(2)},0]$			
7			$z_{LH}^{(2)}[m^{(2)},1]$	$y_H^{(2)}[m^{(2)},1]$			
8			$z_{LL}^{(2)}[m^{(2)},1]$	$y_L^{(2)}[m^{(2)},1]$	$z_{LH}^{(1)}[m^{(1)},0]$	$y_H^{(1)}[m^{(1)},0]$	
9			$z_{LH}^{(2)}[m^{(2)},2]$	$y_H^{(2)}[m^{(2)},2]$	$z_{LL}^{(1)}[m^{(1)},0]$	$y_L^{(1)}[m^{(1)},0]$	
10	$z_{LL}^{(3)}[m^{(3)},2]$	$y_L^{(3)}[m^{(3)},2]$			$z_{LH}^{(1)}[m^{(1)},1]$	$y_H^{(1)}[m^{(1)},1]$	
11			$z_{LL}^{(2)}[m^{(2)},2]$	$y_L^{(2)}[m^{(2)},2]$	$z_{LL}^{(1)}[m^{(1)},1]$	$y_L^{(1)}[m^{(1)},1]$	$w[m,0]$
12					$z_{LH}^{(1)}[m^{(1)},2]$	$y_H^{(1)}[m^{(1)},2]$	$w[m,1]$
13			$z_{LH}^{(2)}[m^{(2)},3]$	$y_H^{(2)}[m^{(2)},3]$	$z_{LL}^{(1)}[m^{(1)},2]$	$y_L^{(1)}[m^{(1)},2]$	$w[m,2]$
14	$z_{LH}^{(3)}[m^{(3)},3]$	$y_H^{(3)}[m^{(3)},3]$			$z_{LH}^{(1)}[m^{(1)},3]$	$y_H^{(1)}[m^{(1)},3]$	$w[m,3]$
15			$z_{LL}^{(2)}[m^{(2)},3]$	$y_L^{(2)}[m^{(2)},3]$	$z_{LL}^{(1)}[m^{(1)},3]$	$y_L^{(1)}[m^{(1)},3]$	$w[m,4]$
16					$z_{LH}^{(1)}[m^{(1)},4]$	$y_H^{(1)}[m^{(1)},4]$	$w[m,5]$
17			$z_{LH}^{(2)}[m^{(2)},4]$	$y_H^{(2)}[m^{(2)},4]$	$z_{LL}^{(1)}[m^{(1)},4]$	$y_L^{(1)}[m^{(1)},4]$	$w[m,6]$
18	$z_{LL}^{(3)}[m^{(3)},3]$	$y_L^{(3)}[m^{(3)},3]$			$z_{LH}^{(1)}[m^{(1)},5]$	$y_H^{(1)}[m^{(1)},5]$	$w[m,7]$
19			$z_{LL}^{(2)}[m^{(2)},4]$	$y_L^{(2)}[m^{(2)},4]$	$z_{LL}^{(1)}[m^{(1)},5]$	$y_L^{(1)}[m^{(1)},5]$	$w[m,8]$
20					$z_{LH}^{(1)}[m^{(1)},6]$	$y_H^{(1)}[m^{(1)},6]$	$w[m,9]$
21			$z_{LH}^{(2)}[m^{(2)},5]$	$y_H^{(2)}[m^{(2)},5]$	$z_{LL}^{(1)}[m^{(1)},6]$	$y_L^{(1)}[m^{(1)},6]$	$w[m,10]$
22					$z_{LH}^{(1)}[m^{(1)},7]$	$y_H^{(1)}[m^{(1)},7]$	$w[m,11]$
23			$z_{LL}^{(2)}[m^{(2)},5]$	$y_L^{(2)}[m^{(2)},5]$	$z_{LH}^{(1)}[m^{(1)},7]$	$y_L^{(1)}[m^{(1)},7]$	$w[m,12]$
					$z_{LH}^{(1)}[m^{(1)},8]$	$y_H^{(1)}[m^{(1)},8]$	$w[m,13]$
					$z_{LL}^{(1)}[m^{(1)},8]$	$y_L^{(1)}[m^{(1)},8]$	$w[m,14]$

Slika 4.13. Vremenski dijagram 2-D inverzne DWT, opisane u ovoj disertaciji, na početku parnih linija

Proces inverznog vertikalnog filtriranja linije počinje čim se komponenta dekomponovanog signala $z_{LH}^{(3)}[m^{(3)},0]$ pojavi na ulazu vertikalnog filtra (u taktu $n=0$ kao što je to naznačeno na slici 4.13). U istom taktu se rezultat vertikalnog filtriranja ($y_H^{(3)}[m^{(3)},0]$) pojavljuje na ulazu horizontalnog filtra iz nivoa 3. U sledećem taktu komponenta signala $z_{LL}^{(3)}[m^{(3)},0]$ se pojavljuje na ulazu vertikalnog filtra. Rezultat

vertikalnog filtriranja ($y_L^{(3)}[m^{(3)},0]$) se pojavljuje na ulazu horizontalnog filtra iz nivoa 3 u istom taktu. U taktu $n=2$ komponenta signala $z_{LH}^{(3)}[m^{(3)},1]$ se pojavljuje na ulazu vertikalnog filtra i rezultat $y_H^{(3)}[m^{(3)},1]$ se generiše i prosleđuje na ulaz horizontalnog filtra iz nivoa 3. U taktu $n=3$ komponenta signala $z_{LL}^{(3)}[m^{(3)},1]$ se pojavljuje na ulazu vertikalnog filtra, rezultat $y_L^{(3)}[m^{(3)},1]$ se generiše i prosleđuje na ulaz horizontalnog filtra iz nivoa 3 i prvi validni odbirak signala generisan ovim horizontalnim filtrom (komponenta signala iz LL podopsega iz nivoa 2: $z_{LL}^{(2)}[m^{(2)},0]$) se pojavljuje na njegovom izlazu. Naredna komponenta signala se pojavljuje na ulazu vertikalnog filtra u taktu $n=8$, dok se ostale komponente signala pojavljuju u svakom četvrtom taktu nakon toga (tj. u taktovima $n=12$, $n=16$, $n=20$, itd.). U ovim taktovima (počev od takta $n=8$), odgovarajuće visokofrekventne komponente signala $y_H^{(3)}[m^{(3)},n^{(3)}]$ i niskofrekventne komponente signala $y_L^{(3)}[m^{(3)},n^{(3)}]$ se naizmenično generišu od strane vertikalnog filtra i prosleđuju na ulaz horizontalnog filtra iz nivoa 3, nakon čega se odgovarajuće rezultujuće komponente signala iz LL podopsega iz nivoa 2 ($z_{LL}^{(2)}[m^{(2)},n^{(2)}]$) generišu na izlazu ovog horizontalnog filtra.

Proces inverznog vertikalnog filtriranja na nivou 2 počinje u taktu $n=4$ kada se komponenta dekomponovanog signala $z_{LH}^{(2)}[m^{(2)},0]$ pojavi na ulazu vertikalnog filtra. U istom taktu se rezultat vertikalnog filtriranja ($y_H^{(2)}[m^{(2)},0]$) pojavljuje na ulazu horizontalnog filtra iz nivoa 2. U sledećem taktu komponenta signala $z_{LL}^{(2)}[m^{(2)},0]$ se pojavljuje na ulazu vertikalnog filtra. Rezultat vertikalnog filtriranja ($y_L^{(2)}[m^{(2)},0]$) se pojavljuje na ulazu horizontalnog filtra iz nivoa 2 u istom taktu. U taktu $n=6$ komponenta signala $z_{LH}^{(2)}[m^{(2)},1]$ se pojavljuje na ulazu vertikalnog filtra i rezultat $y_H^{(2)}[m^{(2)},1]$ se generiše i prosleđuje na ulaz horizontalnog filtra iz nivoa 2. U taktu $n=7$ komponenta signala $z_{LL}^{(2)}[m^{(2)},1]$ se pojavljuje na ulazu vertikalnog filtra, rezultat $y_L^{(2)}[m^{(2)},1]$ se generiše i prosleđuje na ulaz horizontalnog filtra iz nivoa 2 i prvi validni odbirak signala generisan ovim horizontalnim filtrom (komponenta signala iz LL podopsega iz nivoa 1: $z_{LL}^{(1)}[m^{(1)},0]$) se pojavljuje na njegovom izlazu. Naredna komponenta signala se pojavljuje na ulazu vertikalnog filtra u taktu $n=9$, dok se ostale

komponente signala pojavljuju u svakom drugom taktu nakon toga (tj. u taktovima $n=11$, $n=13$, $n=15$, itd.). U ovim taktovima (počev od takta $n=9$), odgovarajuće visokofrekventne komponente signala $y_H^{(2)}[m^{(2)}, n^{(2)}]$ i niskofrekventne komponente signala $y_L^{(2)}[m^{(2)}, n^{(2)}]$ se naizмениčno generišu od strane vertikalnog filtra i prosleđuju na ulaz horizontalnog filtra iz nivoa 2, nakon čega se odgovarajuće rezultujuće komponente signala iz LL podopsega iz nivoa 1 ($z_{LL}^{(1)}[m^{(1)}, n^{(1)}]$) generišu na izlazu ovog horizontalnog filtra.

Proces inverznog vertikalnog filtriranja na nivou 1 počinje u taktu $n=8$. U sukcesivnim ciklusima takta komponente dekomponovanog signala $z_{LH}^{(1)}[m^{(1)}, n^{(1)}]$ i $z_{LL}^{(1)}[m^{(1)}, n^{(1)}]$ se naizмениčno pojavljuju na ulazu vertikalnog filtra iz nivoa 1, odgovarajuće visokofrekventne komponente signala $y_H^{(1)}[m^{(1)}, n^{(1)}]$ i niskofrekventne komponente signala $y_L^{(1)}[m^{(1)}, n^{(1)}]$ se naizмениčno generišu od strane vertikalnog filtra i prosleđuju na ulaz horizontalnog filtra iz nivoa 1. Počev od takta $n=11$, pikseli rekonstruisane slike $w[m, n]$ se sukcesivno generišu na izlazu horizontalnog filtra iz nivoa 1.

Početni ciklus takta za filtriranje prvog odbirka signala na svakom od nivoa se bira na način koji omogućava ispravno učešljavanje vremenskih slotova u kojima se odbirci signala vertikalno filtriraju. Ovaj pristup omogućava korišćenje jednog inverznog vertikalnog filtra za nivo 1 ("Vertikalni filter A") i drugog inverznog vertikalnog filtra za sve ostale nivoe kompozicije ("Vertikalni filter B"), jer je izbegnuto bilo kakvo preklapanje vremenskih slotova kada se "Vertikalni filter B" koristi.

Opisani obrazac 2-D inverznog filtriranja na početku parnih linija se takođe primenjuje i na sve ostale nivoe kompozicije ($j=4, 5, 6$ i 7), koji nisu prikazani na pojednostavljenom dijagramu sa slike 4.13. Inverzno filtriranje inicijalne četiri komponente signala u liniji na nivou j se sprovodi nakon inverznog filtriranja inicijalne četiri komponente signala na nivou $j+1$, ali pre inverznog filtriranja inicijalne četiri komponente signala na nivou $j-1$. Ciklusi takta za filtriranje pete komponente signala na različitim nivoima kompozicije su izabrani na način da se postigne ispravno učešljavanje tih ciklusa takta sa ciljem da se omogući korišćenje istog vertikalnog filtra za nivoe kompozicije od 2 do 7. Sve ostale komponente signala na nivou 4 se filtriraju u

svakom osmom ciklusu takta, sve ostale komponente signala na nivou 5 se filtriraju u svakom šesnaestom ciklusu takta, sve ostale komponente signala na nivou 6 se filtriraju u svakom tridesetdrugom ciklusu takta, itd.

takt	VF ulaz nivo 3	HF ulaz nivo 3	VF ulaz nivo 2	HF ulaz nivo 2	VF ulaz nivo 1	HF ulaz nivo 1	Izlazni pikseli
1896			$z_{HH}^{(2)}[m^{(2)},474]$	$y_H^{(2)}[m^{(2)},474]$			
1897					$z_{HH}^{(1)}[m^{(1)},945]$	$y_H^{(1)}[m^{(1)},945]$	$w[m,1887]$
1898	$z_{HL}^{(3)}[m^{(3)},238]$	$y_L^{(3)}[m^{(3)},238]$			$z_{HL}^{(1)}[m^{(1)},945]$	$y_L^{(1)}[m^{(1)},945]$	$w[m,1888]$
1899			$z_{HH}^{(2)}[m^{(2)},474]$	$y_H^{(2)}[m^{(2)},474]$	$z_{HH}^{(1)}[m^{(1)},946]$	$y_H^{(1)}[m^{(1)},946]$	$w[m,1889]$
1900					$z_{HL}^{(1)}[m^{(1)},946]$	$y_L^{(1)}[m^{(1)},946]$	$w[m,1890]$
1901			$z_{HH}^{(2)}[m^{(2)},475]$	$y_H^{(2)}[m^{(2)},475]$	$z_{HH}^{(1)}[m^{(1)},947]$	$y_H^{(1)}[m^{(1)},947]$	$w[m,1891]$
1902	$z_{HL}^{(3)}[m^{(3)},239]$	$y_L^{(3)}[m^{(3)},239]$			$z_{HL}^{(1)}[m^{(1)},947]$	$y_L^{(1)}[m^{(1)},947]$	$w[m,1892]$
1903			$z_{HH}^{(2)}[m^{(2)},475]$	$y_H^{(2)}[m^{(2)},475]$	$z_{HH}^{(1)}[m^{(1)},948]$	$y_H^{(1)}[m^{(1)},948]$	$w[m,1893]$
1904					$z_{HL}^{(1)}[m^{(1)},948]$	$y_L^{(1)}[m^{(1)},948]$	$w[m,1894]$
1905			$z_{HH}^{(2)}[m^{(2)},476]$	$y_H^{(2)}[m^{(2)},476]$	$z_{HH}^{(1)}[m^{(1)},949]$	$y_H^{(1)}[m^{(1)},949]$	$w[m,1895]$
1906	$z_{HL}^{(3)}[m^{(3)},239]$	$y_L^{(3)}[m^{(3)},239]$			$z_{HL}^{(1)}[m^{(1)},949]$	$y_L^{(1)}[m^{(1)},949]$	$w[m,1896]$
1907			$z_{HH}^{(2)}[m^{(2)},476]$	$y_H^{(2)}[m^{(2)},476]$	$z_{HH}^{(1)}[m^{(1)},950]$	$y_H^{(1)}[m^{(1)},950]$	$w[m,1897]$
1908					$z_{HL}^{(1)}[m^{(1)},950]$	$y_L^{(1)}[m^{(1)},950]$	$w[m,1898]$
1909			$z_{HH}^{(2)}[m^{(2)},477]$	$y_H^{(2)}[m^{(2)},477]$	$z_{HH}^{(1)}[m^{(1)},951]$	$y_H^{(1)}[m^{(1)},951]$	$w[m,1899]$
1910					$z_{HL}^{(1)}[m^{(1)},951]$	$y_L^{(1)}[m^{(1)},951]$	$w[m,1900]$
1911			$z_{HH}^{(2)}[m^{(2)},477]$	$y_H^{(2)}[m^{(2)},477]$	$z_{HH}^{(1)}[m^{(1)},952]$	$y_H^{(1)}[m^{(1)},952]$	$w[m,1901]$
1912					$z_{HL}^{(1)}[m^{(1)},952]$	$y_L^{(1)}[m^{(1)},952]$	$w[m,1902]$
1913			$z_{HH}^{(2)}[m^{(2)},478]$	$y_H^{(2)}[m^{(2)},478]$	$z_{HH}^{(1)}[m^{(1)},953]$	$y_H^{(1)}[m^{(1)},953]$	$w[m,1903]$
1914					$z_{HL}^{(1)}[m^{(1)},953]$	$y_L^{(1)}[m^{(1)},953]$	$w[m,1904]$
1915			$z_{HH}^{(2)}[m^{(2)},478]$	$y_H^{(2)}[m^{(2)},478]$	$z_{HH}^{(1)}[m^{(1)},954]$	$y_H^{(1)}[m^{(1)},954]$	$w[m,1905]$
1916					$z_{HL}^{(1)}[m^{(1)},954]$	$y_L^{(1)}[m^{(1)},954]$	$w[m,1906]$
1917			$z_{HH}^{(2)}[m^{(2)},479]$	$y_H^{(2)}[m^{(2)},479]$	$z_{HH}^{(1)}[m^{(1)},955]$	$y_H^{(1)}[m^{(1)},955]$	$w[m,1907]$
1918					$z_{HL}^{(1)}[m^{(1)},955]$	$y_L^{(1)}[m^{(1)},955]$	$w[m,1908]$
1919			$z_{HH}^{(2)}[m^{(2)},479]$	$y_H^{(2)}[m^{(2)},479]$	$z_{HH}^{(1)}[m^{(1)},956]$	$y_H^{(1)}[m^{(1)},956]$	$w[m,1909]$
1920					$z_{HL}^{(1)}[m^{(1)},956]$	$y_L^{(1)}[m^{(1)},956]$	$w[m,1910]$
1921					$z_{HH}^{(1)}[m^{(1)},957]$	$y_H^{(1)}[m^{(1)},957]$	$w[m,1911]$
1922					$z_{HL}^{(1)}[m^{(1)},957]$	$y_L^{(1)}[m^{(1)},957]$	$w[m,1912]$
1923					$z_{HH}^{(1)}[m^{(1)},958]$	$y_H^{(1)}[m^{(1)},958]$	$w[m,1913]$
1924					$z_{HL}^{(1)}[m^{(1)},958]$	$y_L^{(1)}[m^{(1)},958]$	$w[m,1914]$
1925					$z_{HH}^{(1)}[m^{(1)},959]$	$y_H^{(1)}[m^{(1)},959]$	$w[m,1915]$
1926					$z_{HL}^{(1)}[m^{(1)},959]$	$y_L^{(1)}[m^{(1)},959]$	$w[m,1916]$
1927							$w[m,1917]$
							$w[m,1918]$
							$w[m,1919]$

Slika 4.14. Vremenski dijagram 2-D inverzne DWT, opisane u ovoj disertaciji, na kraju parnih linija

Vremenski dijagram koji ilustruje dinamiku 2-D inverznog filtriranja na kraju parnih linija slike HD rezolucije, za linije čiji početak je prikazan na slici 4.13, predstavljen je na slici 4.14. Već opisani obrazac filtriranja, počev od pete komponente signala na svakom od nivoa, nastavlja se sve do zadnje komponente signala unutar linije za svaki od nivoa. Zadnji piksel rekonstruisane slike ($w[m,1919]$) se generiše tri ciklusa takta nakon filtriranja zadnje komponente signala unutar linije na nivou 1.

Vremenski dijagram koji ilustruje dinamiku 2-D inverznog filtriranja na početku neparnih linija (brojeći od 0) je predstavljen na slici 4.15a, dok je vremenski dijagram koji ilustruje dinamiku 2-D inverznog filtriranja na kraju neparnih linija za slike HD rezolucije predstavljen na slici 4.15b. Obrazac inverznog filtriranja u ovom slučaju je skoro isti kao i u slučaju inverznog filtriranja parnih linija. Postoje samo dve razlike. Prva razlika je u tome što svaka parna komponenta signala (brojeći od 0) koja se pojavljuje na ulazu vertikalnog filtra pripada podopsegu HH ($z_{HH}^{(1)}[m^{(1)}, n^{(1)}]$), dok svaka neparna komponenta signala koja se pojavljuje na ulazu vertikalnog filtra pripada podopsegu HL ($z_{HL}^{(1)}[m^{(1)}, n^{(1)}]$). Druga razlika je u tome što je prvi nivo kompozicije ujedno i jedini nivo kompozicije pošto ni komponente signala iz HH podopsega, niti komponente signala iz HL podopsega nisu generisane na osnovu komponenti signala iz prethodnih nivoa kompozicije.

<i>takt</i>	VF ulaz nivo 1	HF ulaz nivo 1	Izlazni pikseli	<i>takt</i>	VF ulaz nivo 1	HF ulaz nivo 1	Izlazni pikseli
0	$z_{HH}^{(1)}[m^{(1)},0]$	$y_H^{(1)}[m^{(1)},0]$		1912	$z_{HH}^{(1)}[m^{(1)},954]$	$y_H^{(1)}[m^{(1)},954]$	$w[m,1905]$
1	$z_{HL}^{(1)}[m^{(1)},0]$	$y_L^{(1)}[m^{(1)},0]$		1913	$z_{HL}^{(1)}[m^{(1)},954]$	$y_L^{(1)}[m^{(1)},954]$	$w[m,1906]$
2	$z_{HH}^{(1)}[m^{(1)},1]$	$y_H^{(1)}[m^{(1)},1]$		1914	$z_{HH}^{(1)}[m^{(1)},955]$	$y_H^{(1)}[m^{(1)},955]$	$w[m,1907]$
3	$z_{HL}^{(1)}[m^{(1)},1]$	$y_L^{(1)}[m^{(1)},1]$	$w[m,0]$	1915	$z_{HL}^{(1)}[m^{(1)},955]$	$y_L^{(1)}[m^{(1)},955]$	$w[m,1908]$
4	$z_{HH}^{(1)}[m^{(1)},2]$	$y_H^{(1)}[m^{(1)},2]$	$w[m,1]$	1916	$z_{HH}^{(1)}[m^{(1)},956]$	$y_H^{(1)}[m^{(1)},956]$	$w[m,1909]$
5	$z_{HL}^{(1)}[m^{(1)},2]$	$y_L^{(1)}[m^{(1)},2]$	$w[m,2]$	1917	$z_{HL}^{(1)}[m^{(1)},956]$	$y_L^{(1)}[m^{(1)},956]$	$w[m,1910]$
6	$z_{HH}^{(1)}[m^{(1)},3]$	$y_H^{(1)}[m^{(1)},3]$	$w[m,3]$	1918	$z_{HH}^{(1)}[m^{(1)},957]$	$y_H^{(1)}[m^{(1)},957]$	$w[m,1911]$
7	$z_{HL}^{(1)}[m^{(1)},3]$	$y_L^{(1)}[m^{(1)},3]$	$w[m,4]$	1919	$z_{HL}^{(1)}[m^{(1)},957]$	$y_L^{(1)}[m^{(1)},957]$	$w[m,1912]$
8	$z_{HH}^{(1)}[m^{(1)},4]$	$y_H^{(1)}[m^{(1)},4]$	$w[m,5]$	1920	$z_{HH}^{(1)}[m^{(1)},958]$	$y_H^{(1)}[m^{(1)},958]$	$w[m,1913]$
9	$z_{HL}^{(1)}[m^{(1)},4]$	$y_L^{(1)}[m^{(1)},4]$	$w[m,6]$	1921	$z_{HL}^{(1)}[m^{(1)},958]$	$y_L^{(1)}[m^{(1)},958]$	$w[m,1914]$
10	$z_{HH}^{(1)}[m^{(1)},5]$	$y_H^{(1)}[m^{(1)},5]$	$w[m,7]$	1922	$z_{HH}^{(1)}[m^{(1)},959]$	$y_H^{(1)}[m^{(1)},959]$	$w[m,1915]$
11	$z_{HL}^{(1)}[m^{(1)},5]$	$y_L^{(1)}[m^{(1)},5]$	$w[m,8]$	1923	$z_{HL}^{(1)}[m^{(1)},959]$	$y_L^{(1)}[m^{(1)},959]$	$w[m,1916]$
12	$z_{HH}^{(1)}[m^{(1)},6]$	$y_H^{(1)}[m^{(1)},6]$	$w[m,9]$	1924			$w[m,1917]$
13	$z_{HL}^{(1)}[m^{(1)},6]$	$y_L^{(1)}[m^{(1)},6]$	$w[m,10]$	1925			$w[m,1918]$
14				1926			$w[m,1919]$

(a)

(b)

Slika 4.15. Vremenski dijagram 2-D inverzne DWT, opisane u ovoj disertaciji, na početku neparnih linija (a) i na kraju neparnih linija (b)

Ilustracija početka filtriranja po linijama u 2-D inverznoj DWT arhitekturi, opisanoj u ovoj disertaciji, je prikazana na slici 4.16. Sa ciljem pojednostavljenja vremenskog dijagrama, prikazana su samo tri nivoa kompozicije.

Linija 0 komponenti signala iz nivoa 3 na ulazu u “Vertikalni filter B” sadrži komponente signala naizmenično iz podopsega LH i LL, tj. notacija “ $z_{LH}^{(3)}[0, n^{(3)}], z_{LL}^{(3)}[0, n^{(3)}]$ ” sa slike 4.16 zapravo predstavlja sledeću sekvencu komponenti signala $z_{LH}^{(3)}[0,0], z_{LL}^{(3)}[0,0], z_{LH}^{(3)}[0,1], z_{LL}^{(3)}[0,1], z_{LH}^{(3)}[0,2], z_{LL}^{(3)}[0,2]$, itd. Ove komponente signala se vertikalno filtriraju pomoću “Vertikalnog filtra B”, a zatim se horizontalno filtriraju pomoću “Horizontalnog filtra nivoa 3”. Linija 1 komponenti signala iz nivoa 3 na ulazu u “Vertikalni filter B” sadrži komponente signala naizmenično iz podopsega HH i HL, tj. notacija “ $z_{HH}^{(3)}[1, n^{(3)}], z_{HL}^{(3)}[1, n^{(3)}]$ ” sa slike 4.16 zapravo predstavlja sledeću sekvencu komponenti signala $z_{HH}^{(3)}[1,0], z_{HL}^{(3)}[1,0], z_{HH}^{(3)}[1,1], z_{HL}^{(3)}[1,1], z_{HH}^{(3)}[1,2], z_{HL}^{(3)}[1,2]$, itd. Ove komponente signala se takođe vertikalno filtriraju pomoću “Vertikalnog filtra B”, a zatim se horizontalno filtriraju pomoću “Horizontalnog filtra nivoa 3”. Linija 2 komponenti signala iz nivoa 3 na ulazu u “Vertikalni filter B” sadrži komponente signala naizmenično iz podopsega LH i LL ($z_{LH}^{(3)}[2, n^{(3)}], z_{LL}^{(3)}[2, n^{(3)}]$). Nakon vertikalnog filtriranja ove linije pomoću “Vertikalnog filtra B”, a zatim i horizontalnog filtriranja pomoću “Horizontalnog filtra nivoa 3”, generišu se komponente signala $z_{LL}^{(2)}[0, n^{(2)}]$ iz podopsega LL iz nivoa 2. Sledeća linija komponenti signala iz nivoa 3, koja sadrži komponente signala naizmenično iz podopsega HH i HL, pojavljuje se na ulazu “Vertikalnog filtra B” nakon praznog vremenskog intervala u trajanju od jedne linije. Za sve ostale linije iz nivoa 3, sledeći obrazac se nastavlja: linije sa komponentama signala iz podopsega LH i LL i linije sa komponentama signala iz podopsega HH i HL se naizmenično pojavljuju na ulazu “Vertikalnog filtra B”, sa praznim vremenskim intervalom u trajanju od tri linije između sukcesivnih linija. Sve ove linije se vertikalno filtriraju “Vertikalnim filtrom B”, a zatim i horizontalno filtriraju “Horizontalnim filtrom nivoa 3”, generišući komponente signala iz podopsega LL iz nivoa 2 ($z_{LL}^{(2)}[m^{(2)}, n^{(2)}]$), koje će se kasnije koristiti za inverzno filtriranje na nivou 2.

Ulazne linije	VF ulaz nivo 3	VF ulaz nivo 2	VF ulaz nivo 1	Izlazne linije
0	$z_{LH}^{(3)}[0, n^{(3)}], z_{LL}^{(3)}[0, n^{(3)}]$			
1	$z_{HH}^{(3)}[1, n^{(3)}], z_{HL}^{(3)}[1, n^{(3)}]$			
2	$z_{LH}^{(3)}[2, n^{(3)}], z_{LL}^{(3)}[2, n^{(3)}]$	$z_{LH}^{(2)}[0, n^{(2)}], z_{LL}^{(2)}[0, n^{(2)}]$		
3		$z_{HH}^{(2)}[1, n^{(2)}], z_{HL}^{(2)}[1, n^{(2)}]$		
4	$z_{HH}^{(3)}[3, n^{(3)}], z_{HL}^{(3)}[3, n^{(3)}]$	$z_{LH}^{(2)}[2, n^{(2)}], z_{LL}^{(2)}[2, n^{(2)}]$	$z_{LH}^{(1)}[0, n^{(1)}], z_{LL}^{(1)}[0, n^{(1)}]$	
5			$z_{HH}^{(1)}[1, n^{(1)}], z_{HL}^{(1)}[1, n^{(1)}]$	
6		$z_{HH}^{(2)}[3, n^{(2)}], z_{HL}^{(2)}[3, n^{(2)}]$	$z_{LH}^{(1)}[2, n^{(1)}], z_{LL}^{(1)}[2, n^{(1)}]$	linija 0
7			$z_{HH}^{(1)}[3, n^{(1)}], z_{HL}^{(1)}[3, n^{(1)}]$	linija 1
8	$z_{LH}^{(3)}[4, n^{(3)}], z_{LL}^{(3)}[4, n^{(3)}]$	$z_{LH}^{(2)}[4, n^{(2)}], z_{LL}^{(2)}[4, n^{(2)}]$	$z_{LH}^{(1)}[4, n^{(1)}], z_{LL}^{(1)}[4, n^{(1)}]$	linija 2
9			$z_{HH}^{(1)}[5, n^{(1)}], z_{HL}^{(1)}[5, n^{(1)}]$	linija 3
10		$z_{HH}^{(2)}[5, n^{(2)}], z_{HL}^{(2)}[5, n^{(2)}]$	$z_{LH}^{(1)}[6, n^{(1)}], z_{LL}^{(1)}[6, n^{(1)}]$	linija 4
11			$z_{HH}^{(1)}[7, n^{(1)}], z_{HL}^{(1)}[7, n^{(1)}]$	linija 5
12	$z_{LH}^{(3)}[5, n^{(3)}], z_{LL}^{(3)}[5, n^{(3)}]$	$z_{LH}^{(2)}[6, n^{(2)}], z_{LL}^{(2)}[6, n^{(2)}]$	$z_{LH}^{(1)}[8, n^{(1)}], z_{LL}^{(1)}[8, n^{(1)}]$	linija 6
13			$z_{HH}^{(1)}[9, n^{(1)}], z_{HL}^{(1)}[9, n^{(1)}]$	linija 7
14		$z_{HH}^{(2)}[7, n^{(2)}], z_{HL}^{(2)}[7, n^{(2)}]$	$z_{LH}^{(1)}[10, n^{(1)}], z_{LL}^{(1)}[10, n^{(1)}]$	linija 8
15			$z_{HH}^{(1)}[11, n^{(1)}], z_{HL}^{(1)}[11, n^{(1)}]$	linija 9
16	$z_{LH}^{(3)}[6, n^{(3)}], z_{LL}^{(3)}[6, n^{(3)}]$	$z_{LH}^{(2)}[8, n^{(2)}], z_{LL}^{(2)}[8, n^{(2)}]$	$z_{LH}^{(1)}[12, n^{(1)}], z_{LL}^{(1)}[12, n^{(1)}]$	linija 10

Slika 4.16. Početak inverznog filtriranja po linijama

Filtriranje linije 0 iz nivoa 2 je učešljano sa filtriranjem linije 2 iz nivoa 3. Obrazac filtriranja po linijama na nivou 2 je skoro isti kao i na nivou 3. Jedina razlika je u tome što se sukcesivne linije, počev od linije 2, pojavljuju na ulazu “Vertikalnog filtra B” sa praznim vremenskim intervalom u trajanju od jedne linije između sukcesivnih linija. Sve ove linije se vertikalno filtriraju “Vertikalnim filtrom B”, a zatim i horizontalno filtriraju “Horizontalnim filtrom nivoa 2”, generišući komponente signala iz podopsega LL iz nivoa 1 ($z_{LL}^{(1)}[m^{(1)}, n^{(1)}]$), koje će se kasnije koristiti za inverzno filtriranje na nivou 1.

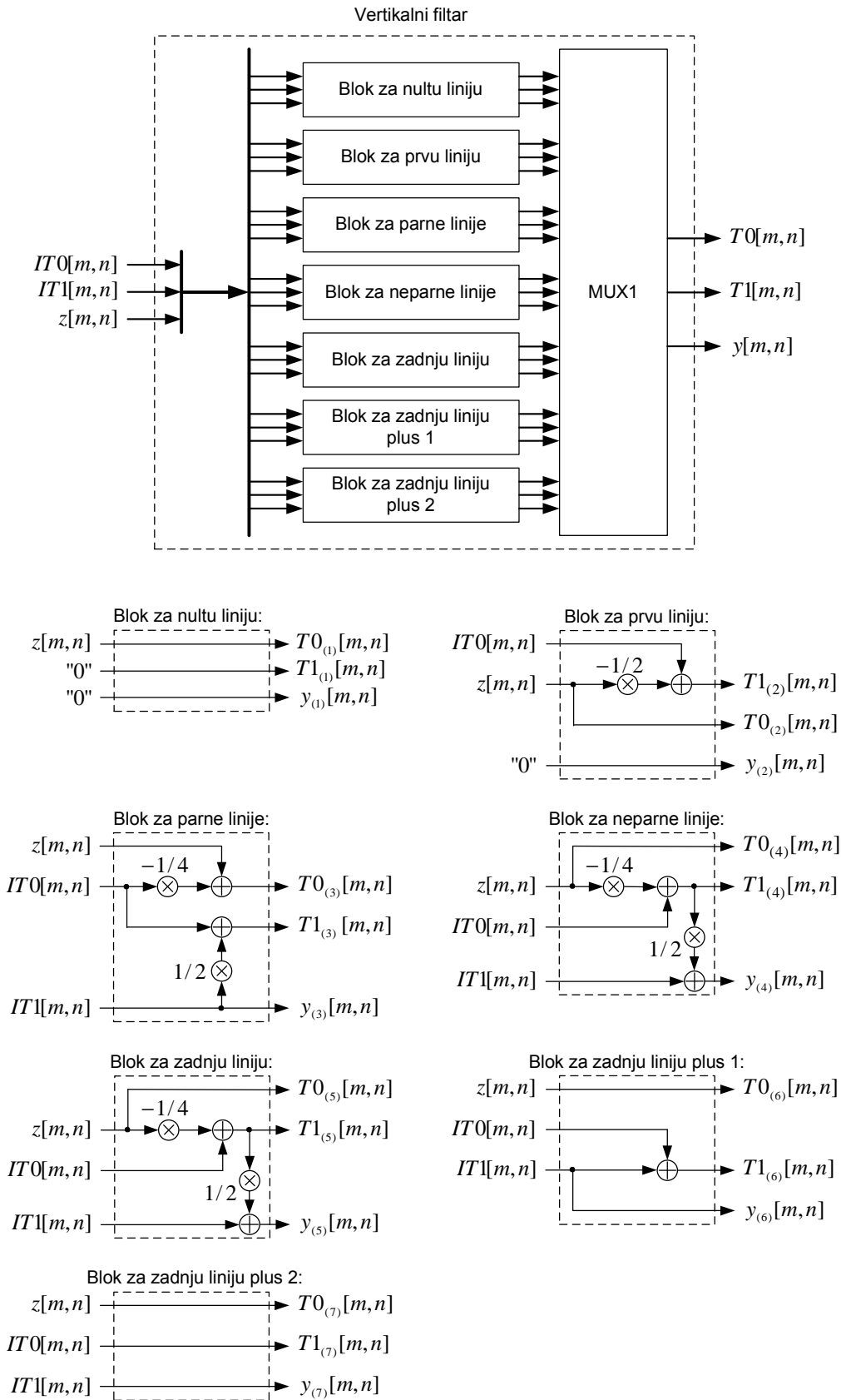
Filtriranje linije 0 iz nivoa 1 je učešljano sa filtriranjem linije 2 iz nivoa 2. Sve linije iz nivoa 1 se pojavljuju na ulazu “Vertikalnog filtra A” sukcesivno, jedna za drugom, bez praznih vremenskih intervala između sukcesivnih linija. Sve ove linije se vertikalno filtriraju “Vertikalnim filtrom A”, a zatim i horizontalno filtriraju “Horizontalnim filtrom nivoa 1”, generišući linije rekonstruisane slike. Linija 0 rekonstruisane slike je učešljana sa filtriranjem linije 2 iz nivoa 1.

Ulazne linije	VF ulaz nivo 3	VF ulaz nivo 2	VF ulaz nivo 1	Izlazne linije
1064	$z_{LH}^{(3)}[268, n^{(3)}], z_{HL}^{(3)}[268, n^{(3)}]$	$z_{LH}^{(2)}[532, n^{(2)}], z_{HL}^{(2)}[532, n^{(2)}]$	$z_{LH}^{(1)}[1060, n^{(1)}], z_{HL}^{(1)}[1060, n^{(1)}]$	linija 1058
1065			$z_{LH}^{(1)}[1061, n^{(1)}], z_{HL}^{(1)}[1061, n^{(1)}]$	linija 1059
1066		$z_{HH}^{(2)}[533, n^{(2)}], z_{HL}^{(2)}[533, n^{(2)}]$	$z_{LH}^{(1)}[1062, n^{(1)}], z_{HL}^{(1)}[1062, n^{(1)}]$	linija 1060
1067			$z_{LH}^{(1)}[1063, n^{(1)}], z_{HL}^{(1)}[1063, n^{(1)}]$	linija 1061
1068	$z_{HH}^{(3)}[269, n^{(3)}], z_{HL}^{(3)}[269, n^{(3)}]$	$z_{LH}^{(2)}[534, n^{(2)}], z_{HL}^{(2)}[534, n^{(2)}]$	$z_{LH}^{(1)}[1064, n^{(1)}], z_{HL}^{(1)}[1064, n^{(1)}]$	linija 1062
1069			$z_{LH}^{(1)}[1065, n^{(1)}], z_{HL}^{(1)}[1065, n^{(1)}]$	linija 1063
1070		$z_{HH}^{(2)}[535, n^{(2)}], z_{HL}^{(2)}[535, n^{(2)}]$	$z_{LH}^{(1)}[1066, n^{(1)}], z_{HL}^{(1)}[1066, n^{(1)}]$	linija 1064
1071			$z_{LH}^{(1)}[1067, n^{(1)}], z_{HL}^{(1)}[1067, n^{(1)}]$	linija 1065
1072	privremeni rezultat 1 ⁽³⁾	$z_{LH}^{(2)}[536, n^{(2)}], z_{HL}^{(2)}[536, n^{(2)}]$	$z_{LH}^{(1)}[1068, n^{(1)}], z_{HL}^{(1)}[1068, n^{(1)}]$	linija 1066
1073			$z_{HH}^{(1)}[1069, n^{(1)}], z_{HL}^{(1)}[1069, n^{(1)}]$	linija 1067
1074		$z_{HH}^{(2)}[537, n^{(2)}], z_{HL}^{(2)}[537, n^{(2)}]$	$z_{LH}^{(1)}[1070, n^{(1)}], z_{HL}^{(1)}[1070, n^{(1)}]$	linija 1068
1075			$z_{LH}^{(1)}[1071, n^{(1)}], z_{HL}^{(1)}[1071, n^{(1)}]$	linija 1069
1076	privremeni rezultat 2 ⁽³⁾	$z_{LH}^{(2)}[538, n^{(2)}], z_{HL}^{(2)}[538, n^{(2)}]$	$z_{LH}^{(1)}[1072, n^{(1)}], z_{HL}^{(1)}[1072, n^{(1)}]$	linija 1070
1077			$z_{LH}^{(1)}[1073, n^{(1)}], z_{HL}^{(1)}[1073, n^{(1)}]$	linija 1071
1078		$z_{HH}^{(2)}[539, n^{(2)}], z_{HL}^{(2)}[539, n^{(2)}]$	$z_{LH}^{(1)}[1074, n^{(1)}], z_{HL}^{(1)}[1074, n^{(1)}]$	linija 1072
1079		privremeni rezultat 1 ⁽²⁾	$z_{LH}^{(1)}[1075, n^{(1)}], z_{HL}^{(1)}[1075, n^{(1)}]$	linija 1073
		privremeni rezultat 2 ⁽²⁾	$z_{LH}^{(1)}[1076, n^{(1)}], z_{HL}^{(1)}[1076, n^{(1)}]$	linija 1074
			$z_{HH}^{(1)}[1077, n^{(1)}], z_{HL}^{(1)}[1077, n^{(1)}]$	linija 1075
			$z_{LH}^{(1)}[1078, n^{(1)}], z_{HL}^{(1)}[1078, n^{(1)}]$	linija 1076
			$z_{HH}^{(1)}[1079, n^{(1)}], z_{HL}^{(1)}[1079, n^{(1)}]$	linija 1077
			privremeni rezultat 1 ⁽¹⁾	linija 1078
			privremeni rezultat 2 ⁽¹⁾	linija 1079

Slika 4.17. Kraj inverznog filtriranja po linijama

Ilustracija kraja filtriranja po linijama za slike HD rezolucije u 2-D inverznoj DWT arhitekturi, opisanoj u ovoj disertaciji, je prikazana na slici 4.17. Već opisani obrazac filtriranja po linijama se nastavlja do zadnje linije komponenti signala na svakom od nivoa. Nakon što se sve linije sa komponentama signala isfiltriraju pomoću odgovarajućeg inverznog vertikalnog filtra i odgovarajućeg inverznog horizontalnog filtra, u poslednja dva vremenska intervala u trajanju od jedne linije po intervalu “Vertikalni filter B” ili “Vertikalni filter A” se koriste za procesiranje internih međurezultata “privremeni rezultat 1” i “privremeni rezultat 2”. Ovi interni međurezultati iz nivoa j se koriste za generisanje zadnje dve linije rezultujućih komponenti signala iz podopsega LL iz nivoa $j-1$ ($j > 1$). Slično tome, dve linije internih međurezultata iz nivoa 1 se koriste za generisanje zadnje dve linije rekonstruisane slike.

Interna struktura “Vertikalnog filtra A” i “Vertikalnog filtra B” sa slike 4.12, prikazana je na slici 4.18.



Slika 4.18. Interna struktura inverznog vertikalnog filtra

Relacije koje povezuju ulazne i izlazne signale “Blok za nultu liniju” su opisane sledećim jednačinama:

$$\begin{aligned} T0[m, n] &= z[m, n], \\ T1[m, n] &= 0, \\ y[m, n] &= 0. \end{aligned} \tag{4.29}$$

Ovaj blok prima odbirke signala iz linije 0 kao ulazni signal $z[m, n]$, dok na izlazu $y[m, n]$ generiše set nula.

Zavisnosti između ulaznih i izlaznih signala za “Blok za prvu liniju” su opisane jednačinama (4.30):

$$\begin{aligned} T0[m, n] &= z[m, n], \\ T1[m, n] &= IT0[m, n] - \frac{1}{2} z[m, n] = T0[m-1, n] - \frac{1}{2} z[m, n] \\ &= -\frac{1}{2} z[m, n] + z[m-1, n], \\ y[m, n] &= 0. \end{aligned} \tag{4.30}$$

Ovaj blok prima odbirke signala iz linije 1 kao ulazni signal $z[m, n]$, dok na izlazu $y[m, n]$ generiše set nula.

Jednačine (4.31) opisuju relacije između ulaznih i izlaznih signala za “Blok za parne linije” kada on prima odbirke signala iz linije 2 kao ulazni signal $z[m, n]$. Izlazni signal ovog bloka $y[m, n]$ je opisan jednačinom koja odgovara specijalnoj formi niskopropusnog *Le Gall*-ovog 5/3 inverznog filtra, koja se koristi za inverzno vertikalno filtriranje na granicama slike, umesto dobro poznatog postupka simetričnog proširenja sekvence odbiraka signala na granicama slike. Ovaj izlazni signal $y[m, n]$ zapravo predstavlja liniju 0 rezultujućih validnih odbiraka signala generisanih inverznim vertikalnim filtrom.

$$\begin{aligned}
T0[m, n] &= z[m, n] - \frac{1}{4} ITO[m, n] = z[m, n] - \frac{1}{4} T0[m-1, n] \\
&= z[m, n] - \frac{1}{4} z[m-1, n], \\
T1[m, n] &= ITO[m, n] + \frac{1}{2} IT1[m, n] = T0[m-1, n] + \frac{1}{2} T1[m-1, n] \quad (4.31) \\
&= \frac{3}{4} z[m-1, n] + \frac{1}{2} z[m-2, n], \\
y[m, n] &= IT1[m, n] = T1[m-1, n] = -\frac{1}{2} z[m-1, n] + z[m-2, n].
\end{aligned}$$

Jednačine (4.32) opisuju relacije između ulaznih i izlaznih signala za “Blok za neparne linije” kada on prima odbirke signala iz linije 3 kao ulazni signal $z[m, n]$. Izlazni signal ovog bloka $y[m, n]$ je opisan jednačinom koja odgovara specijalnoj formi visokopropusnog *Le Gall*-ovog 5/3 inverznog filtra, koja se koristi za inverzno vertikalno filtriranje na granicama slike, umesto postupka simetričnog proširenja sekvence odbiraka signala na granicama slike. Ovaj izlazni signal $y[m, n]$ zapravo predstavlja liniju 1 rezultujućih validnih odbiraka signala generisanih inverznim vertikalnim filtrom.

$$\begin{aligned}
T0[m, n] &= z[m, n], \\
T1[m, n] &= ITO[m, n] - \frac{1}{4} z[m, n] = T0[m-1, n] - \frac{1}{4} z[m, n] \\
&= -\frac{1}{4} z[m, n] + z[m-1, n] - \frac{1}{4} z[m-2, n], \\
y[m, n] &= IT1[m, n] + \frac{1}{2} ITO[m, n] - \frac{1}{8} z[m, n] \quad (4.32) \\
&= T1[m-1, n] + \frac{1}{2} T0[m-1, n] - \frac{1}{8} z[m, n] \\
&= -\frac{1}{8} z[m, n] + \frac{1}{2} z[m-1, n] + \frac{5}{8} z[m-2, n] + \frac{1}{2} z[m-3, n].
\end{aligned}$$

Jednačine (4.33) opisuju relacije između ulaznih i izlaznih signala za “Blok za parne linije” kada on prima odbirke signala iz bilo koje parne linije, osim ulazne linije 0 i ulazne linije 2, kao ulazni signal $z[m, n]$. Izlazni signal ovog bloka $y[m, n]$ je opisan jednačinom koja odgovara niskopropusnom *Le Gall*-ovom 5/3 inverznom filtru. Ovaj

izlazni signal $y[m,n]$ zapravo predstavlja bilo koju parnu liniju (brojeći od 0) rezultujućih validnih odbiraka signala generisanih inverznim vertikalnim filtrom, osim linije 0 i parne linije među zadnje tri linije.

$$\begin{aligned}
TO[m,n] &= z[m,n] - \frac{1}{4} ITO[m,n] = z[m,n] - \frac{1}{4} TO[m-1,n] \\
&= z[m,n] - \frac{1}{4} z[m-1,n], \\
TI[m,n] &= ITO[m,n] + \frac{1}{2} ITI[m,n] = TO[m-1,n] + \frac{1}{2} TI[m-1,n] \\
&= \frac{7}{8} z[m-1,n] + \frac{1}{2} z[m-2,n] - \frac{1}{8} z[m-3,n], \\
y[m,n] &= ITI[m,n] = TI[m-1,n] \\
&= -\frac{1}{4} z[m-1,n] + z[m-2,n] - \frac{1}{4} z[m-3,n].
\end{aligned} \tag{4.33}$$

Jednačine (4.34) opisuju relacije između ulaznih i izlaznih signala za “Blok za neparne linije” kada on prima odbirke signala iz bilo koje neparne linije, osim ulazne linije 1, ulazne linije 3 i zadnje ulazne linije, kao ulazni signal $z[m,n]$. Izlazni signal ovog bloka $y[m,n]$, opisan je jednačinom koja odgovara visokopropusnom *Le Gall*-ovom 5/3 inverznom filtru. Ovaj izlazni signal $y[m,n]$ zapravo predstavlja bilo koju neparnu liniju (brojeći od 0) rezultujućih validnih odbiraka signala generisanih inverznim vertikalnim filtrom, osim linije 1 i neparnih linija među zadnje tri linije.

$$\begin{aligned}
TO[m,n] &= z[m,n], \\
TI[m,n] &= ITO[m,n] - \frac{1}{4} z[m,n] = TO[m-1,n] - \frac{1}{4} z[m,n] \\
&= -\frac{1}{4} z[m,n] + z[m-1,n] - \frac{1}{4} z[m-2,n], \\
y[m,n] &= ITI[m,n] + \frac{1}{2} ITO[m,n] - \frac{1}{8} z[m,n] \\
&= TI[m-1,n] + \frac{1}{2} TO[m-1,n] - \frac{1}{8} z[m,n] \\
&= -\frac{1}{8} z[m,n] + \frac{1}{2} z[m-1,n] + \frac{3}{4} z[m-2,n] + \frac{1}{2} z[m-3,n] - \frac{1}{8} z[m-4,n].
\end{aligned} \tag{4.34}$$

Opis preostala tri bloka koja sačinjavaju inverzni vertikalni filter je dat za slučaj kada je ukupan broj linija unutar slike paran.

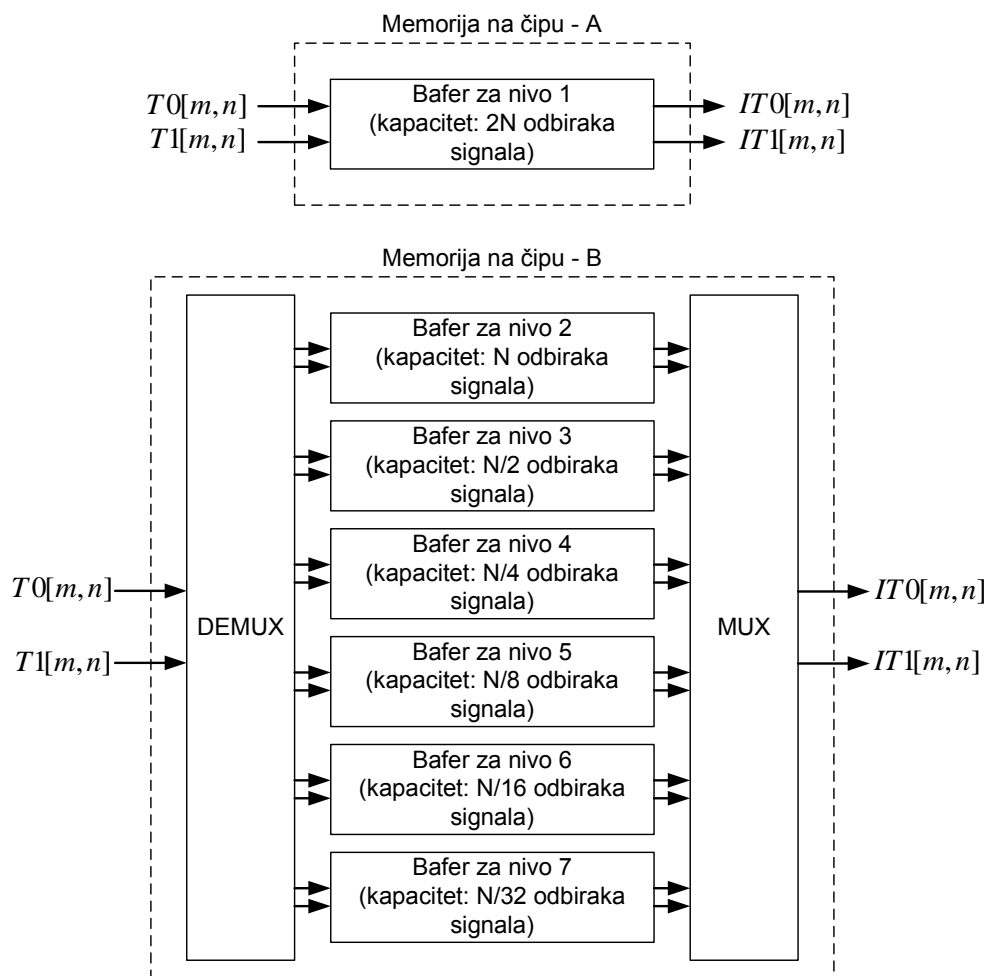
Jednačine (4.35) opisuju relacije između ulaznih i izlaznih signala za “Blok za zadnju liniju” koji prima odbirke signala iz zadnje linije kao ulazni signal $z[m, n]$. Izlazni signal ovog bloka $y[m, n]$, opisan je jednačinom koja odgovara visokopropusnom *Le Gall*-ovom 5/3 inverznom filteru. Jednačine (4.35) su izvedene imajući u vidu činjenicu da parna linija prethodi liniji koju procesira “Blok za zadnju liniju”.

$$\begin{aligned}
 T0[m, n] &= z[m, n], \\
 T1[m, n] &= ITO[m, n] - \frac{1}{4} z[m, n] = T0[m-1, n] - \frac{1}{4} z[m, n] \\
 &= -\frac{1}{4} z[m, n] + z[m-1, n] - \frac{1}{4} z[m-2, n], \\
 y[m, n] &= IT1[m, n] + \frac{1}{2} ITO[m, n] - \frac{1}{8} z[m, n] \\
 &= T1[m-1, n] + \frac{1}{2} T0[m-1, n] - \frac{1}{8} z[m, n] \\
 &= -\frac{1}{8} z[m, n] + \frac{1}{2} z[m-1, n] + \frac{3}{4} z[m-2, n] + \frac{1}{2} z[m-3, n] - \frac{1}{8} z[m-4, n].
 \end{aligned} \tag{4.35}$$

Jednačine (4.36) opisuju relacije između ulaznih i izlaznih signala za “Blok za zadnju liniju plus 1” koji je odgovoran za inverzno vertikalno filtriranje preostalih internih međurezultata $ITO[m, n]$ i $IT1[m, n]$. Izlazni signal ovog bloka $y[m, n]$ je opisan jednačinom koja odgovara niskopropusnom *Le Gall*-ovom 5/3 inverznom filteru.

$$\begin{aligned}
 T0[m, n] &= z[m, n], \\
 T1[m, n] &= ITO[m, n] + IT1[m, n] = T0[m-1, n] + T1[m-1, n] \\
 &= \frac{3}{4} z[m-1, n] + z[m-2, n] - \frac{1}{4} z[m-3, n], \\
 y[m, n] &= IT1[m, n] = T1[m-1, n] \\
 &= -\frac{1}{4} z[m-1, n] + z[m-2, n] - \frac{1}{4} z[m-3, n].
 \end{aligned} \tag{4.36}$$

$$\begin{aligned}
T0[m, n] &= z[m, n], \\
T1[m, n] &= ITO[m, n] = T0[m-1, n] = z[m-1, n], \\
y[m, n] &= IT1[m, n] = T1[m-1, n] \\
&= \frac{3}{4} z[m-2, n] + z[m-3, n] - \frac{1}{4} z[m-4, n].
\end{aligned}
\tag{4.37}$$



Slika 4.19. Memorija na čipu

Konačno, jednačine (4.37) opisuju relacije između ulaznih i izlaznih signala za “Blok za zadnju liniju plus 2” koji je odgovoran za inverzno vertikalno filtriranje preostalih internih međurezultata $IT1[m, n]$. Izlazni signal ovog bloka $y[m, n]$ je opisan jednačinom koja odgovara specijalnoj formi visokopropusnog *Le Gall*-ovog 5/3 inverznog filtra, koja se koristi za inverzno vertikalno filtriranje na granicama slike, umesto postupka simetričnog proširenja sekvence odbiraka signala na granicama slike.

Ovaj izlazni signal $y[m,n]$ zapravo predstavlja zadnju liniju rezultujućih validnih odbiraka signala generisanih inverznim vertikalnim filtrom.

Jednačine (4.30)-(4.37) su izvedene imajući u vidu činjenicu da se za smeštanje internih međurezultata $T0[m,n]$ i $T1[m,n]$ koristi memorija na čipu koja proizvodi sledeće relacije:

$$\begin{aligned} ITO[m,n] &= T0[m-1,n], \\ IT1[m,n] &= T1[m-1,n]. \end{aligned} \quad (4.38)$$

Memorija na čipu, koja se koristi u okviru hardverske realizacije 2-D inverzne DWT opisane u ovoj disertaciji, je prikazana na slici 4.19. Za ispravno inverzno filtriranje i kompoziciju digitalne slike dimenzija $N \times N$ piksela, potrebno je da se po dve linije internih međurezultata filtriranja, za svaki od nivoa kompozicije, smeštaju u memoriju na čipu. Interni međurezultati iz nivoa 1 kompozicije se smeštaju u “Memoriju na čipu - A”, koja sadrži jedan bafer kapaciteta $2N$ odbiraka signala. Interni međurezultati iz ostalih nivoa kompozicije se smeštaju u “Memoriju na čipu - B”, koja sadrži šest bafera (za slučaj $J = 7$ nivoa kompozicije) sa kapacitetom koji se dvostruko smanjuje na svakom narednom nivou kompozicije, počevši od kapaciteta od N odbiraka signala na nivou 2. Svi ovi baferi zapravo predstavljaju FIFO memorije.

4.3.1. Poređenja performansi različitih hardverskih realizacija dvodimenzionalnog inverznog DWT 5/3 filtra

Hardverska realizacija 2-D inverznog DWT 5/3 filtra, opisana u ovoj disertaciji, koristi J FIFO bafera za smeštanje internih međurezultata $T0[m,n]$ i $T1[m,n]$, za J nivoa kompozicije digitalne slike dimenzija $N \times N$. Kapacitet FIFO bafera za nivo 1 je $2N$ odbiraka signala, dok je kapacitet FIFO bafera za svaki naredni nivo kompozicije dvostruko manji od kapaciteta FIFO bafera sa prethodnog nivoa. Osim toga, svaki od nivoa kompozicije zahteva korišćenje četiri ulazna registra za smeštanje komponenti signala iz LL, LH, HL i HH podopsega, dok svaki od korišćenih inverznih horizontalnih

filtera u sebi sadrži još po dva registra (tj. elementa za kašnjenje). Stoga, ukupni kapacitet memorije na čipu koju koristi 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji se može izračunati na sledeći način:

$$2N + N + \frac{N}{2} + \frac{N}{4} + \dots + \frac{N}{2^{J-2}} + 6J = 4N(1 - 2^{-J}) + 6J. \quad (4.39)$$

Hardverska realizacija 2-D inverznog DWT 5/3 filtra, opisana u ovoj disertaciji, ne zahteva korišćenje bilo kakve eksterne memorije. S obzirom da je za kompresiju/dekompresiju bilo koje realne digitalne slike $J \ll N$, ukupni kapacitet korišćene memorije se može aproksimirati sa $4N(1 - 2^{-J})$.

Na osnovu vremenskih dijagrama prikazanih na slikama 4.13 i 4.14, može se izračunati da je vreme potrebno za procesiranje jedne linije $N + 4(J - 1) + 3$ ciklusa takta. Na osnovu vremenskih dijagrama za filtriranje po linijama, prikazanih na slikama 4.16 i 4.17, se može zaključiti da je ukupan broj vremenskih intervala u trajanju od jedne linije, potrebnih za procesiranje svih linija unutar digitalne slike jednak $N + 2J$. Stoga, ukupno vreme potrebno za procesiranje digitalne slike dimenzija $N \times N$ pomoću 2-D inverzne DWT 5/3 arhitekture opisane u ovoj disertaciji se može izraziti kao:

$$(N + 4J - 1) \cdot (N + 2J) \approx N^2. \quad (4.40)$$

Konačno, na osnovu slika 4.13 i 4.16 se može zaključiti da je kašnjenje koje pri procesiranju digitalne slike dimenzija $N \times N$ unosi 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji:

$$2JN + 4(J - 1) + 3 \approx 2JN. \quad (4.41)$$

Efikasnost hardverskog iskorišćenja (*hardware utilization efficiency*) $HUE(PA)$ za paralelne arhitekture (PA) se definiše na sledeći način:

$$HUE(PA) = \frac{\sum_{i=1}^M A_i \times HUE(A_i)}{\sum_{i=1}^M A_i}, \quad (4.42)$$

gde A_i označava i -tu procesnu jedinicu, $HUE(A_i)$ označava efikasnost hardverskog iskorišćenja procesne jedinice A_i , dok M označava ukupan broj procesnih jedinica koje sačinjavaju posmatranu arhitekturu. U ovoj disertaciji opisana 2-D inverzna DWT 5/3 arhitektura se sastoji od dve procesne jedinice: A_1 (koju čini deo arhitekture odgovoran za inverzno filtriranje na nivou 1 kompozicije) i A_2 (koju čini deo arhitekture odgovoran za inverzno filtriranje na svim ostalim nivoima kompozicije j , gde je $j > 1$).

Efikasnost hardverskog iskorišćenja $HUE(A_i)$ za i -tu procesnu jedinicu A_i se može definisati kao odnos broja perioda takta u kojima se vrši izračunavanje vrednosti izlaznih podataka i ukupnog broja perioda takta koliko traje procesiranje podataka. Za 2-D inverznu DWT 5/3 arhitekturu opisanu u ovoj disertaciji, broj perioda takta u kojima se vrši izračunavanje vrednosti izlaznih podataka je jednak broju odbiraka signala koji se generišu na izlazu procesne jedinice A_i .

Stoga se efikasnost hardverskog iskorišćenja za procesnu jedinicu A_1 može izraziti kao:

$$HUE(A_1) = \frac{N \cdot N}{(N + 4) \cdot (N + 2)} \approx 1. \quad (4.43)$$

Efikasnost hardverskog iskorišćenja za procesnu jedinicu A_2 se može izraziti kao:

$$HUE(A_2) = \frac{\frac{1}{3} N^2 (1 - 4^{-J+1})}{(N + 4J - 5) \cdot (N + J - 1)} \approx \frac{1}{3} (1 - 4^{-J+1}). \quad (4.44)$$

Konačno, efikasnost hardverskog iskorišćenja za 2-D inverznu DWT 5/3 arhitekturu opisanu u ovoj disertaciji se može izračunati na sledeći način:

$$HUE(PA) = \frac{1 \cdot 1 + 1 \cdot \frac{1}{3}(1 - 4^{-J+1})}{1 + 1} = \frac{2}{3}(1 - 4^{-J}). \quad (4.45)$$

Dakle, u slučaju kada se vrši inverzno filtriranje i kompozicija na samom jednom nivou kompozicije, efikasnost hardverskog iskorišćenja cele arhitekture opisane u ovoj disertaciji odgovara efikasnosti hardverskog iskorišćenja procesne jedinice A_1 i približno je jednaka 1. Međutim, u slučaju kada se vrši inverzno filtriranje i kompozicija na J ($J > 1$) nivou kompozicije, efikasnost hardverskog iskorišćenja se smanjuje na približno 0,66, što se može izračunati na osnovu jednačine (4.45).

Ukupni kapacitet korišćene memorije je predstavljen u vidu broja odbiraka signala, dok su vreme procesiranja i kašnjenje predstavljeni u vidu broja ciklusa takta.

Tabela 4.3 prikazuje poređenje performansi 2-D inverzne DWT 5/3 arhitekture opisane u ovoj disertaciji sa performansama arhitekture opisanih u [42], [50], [52-54] i [58-76] u pogledu kapaciteta korišćene memorije na čipu, kapaciteta korišćene eksterne memorije, vremena procesiranja i kašnjenja, za J nivou kompozicije digitalne slike dimenzija $N \times N$.

Može se zaključiti da 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji ima vreme procesiranja srednje dužine trajanja i unosi kašnjenje srednje dužine trajanja u poređenju sa ostalim arhitekturama. Efikasnost hardverskog iskorišćenja arhitekture opisane u ovoj disertaciji je uporediva sa ostalim arhitekturama opisanim u literaturi. Međutim, 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji zahteva korišćenje memorije najmanjeg kapaciteta u poređenju sa svim do sada objavljenim 2-D DWT 5/3 arhitekturama. Za $J \rightarrow \infty$ nivou kompozicije digitalne slike dimenzija $N \times N$ 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji zahteva korišćenje memorije ukupnog kapaciteta $4N$ odbiraka signala, što je za 20% manji kapacitet memorije u poređenju sa memorijski najefikasnijom do sada predloženom 2-D DWT 5/3 arhitekturom.

Tabela 4.3. Poređenje performansi različitih 2-D DWT 5/3 arhitektura

Arhitektura	Memorija na čipu	Eksterna memorija	Vreme procesiranja	Kašnjenje	HUE
Neseparabilna [58]	$10N$	0	N^2	n/a	n/a
SIMD [58]	N^2	0	$25J$	n/a	n/a
Direktna [59]	N^2	0	$4N^2$	$2N^2$	n/a
Sistolno-paralelna [59]	$14N$	0	N^2	1	n/a
Paralelna [58,60,61]	$11N$	0	N^2	n/a	n/a
[62]	$10N(1-2^{-J})$	0	$N^2/2$	n/a	$4(1-4^{-J})/3J$
[52]	$7N(1-2^{-J})$	0	$4N^2(1-4^{-J})/3 - 2N(1-2^{-J})$	n/a	n/a
[63]	$10N(1-2^{-J}) - N$	0	$N^2/4$	n/a	$8(1-4^{-J})/9$
[42]	$5N$	$N^2/4$	$2N^2(1-4^{-J})/3$	$2N$	1
[64]	$7N(1-2^{-J})$	0	N^2	n/a	$2(1-4^{-J})/3$
[50]	$N^2 + 4N$	0	$2N^2(1-4^{-J})/3$	$2N$	1
RA [53]	$6N$	0	N^2	$2N$	$2(1-4^{-J})/3$
DSA [54]	$3,5N$	$N^2/4$	$2N^2(1-4^{-J})/3$	n/a	n/a
[65]	$5N$	$N^2/4$	$2N^2(1-4^{-J})/3$	$5N$	1
Konvolucija [66]	$9N$	0	N^2	n/a	n/a
Dizanje [66]	$7N$	0	N^2	n/a	n/a
[67]	$3,5N$	$N^2/4$	$4N^2(1-4^{-J})/3 + 2N(1-2^{-J}) + 10J$	$2N(1-2^{-J})$	1
FA [68]	$3,5N$	$N^2/4$	$N^2(1-4^{-J})/3$	n/a	n/a
PA [68]	$6N(1-2^{-J}) + 0,5N$	0	$N^2/4$	n/a	$8(1-4^{-J})/9$
FA [69]	$3,5N$	$N^2/4$	$2N^2(1-4^{-J})/3$	N	1
HA [69]	$3,5N$	$N^2/4$	$N^2(1-4^{-J})/3$	$N/2$	1
[70]	$\approx 7N$	0	N^2/P	$\approx N/2$	n/a
[71]	$8N(1-2^{-J})$	0	$N^2 + 6N(1-2^{-J})$	$\approx 3N$	0,5-0,6
[72]	$\approx 6,25N$	0	$N^2/16$	n/a	n/a
[73]	$2N$	$N^2/2$	$\approx N^2$	$\approx 3N/2$	n/a
[74]	$2N$	$N^2/4$	$2N^2(1-4^{-J})/3$	N	1
[74]	$3N(1-2^{-J}) + 2N$	0	$\approx N^2/2 + N$	$\approx N$	0,6-0,75
[75,76]	$9N$	0	$N^2/4$	n/a	≈ 1
Ova disertacija	$4N(1-2^{-J})$	0	N^2	$2N$	$2(1-4^{-J})/3$

Legenda: P : veličina ulaznog bloka.

4.3.2. Poređenja rezultata sinteze različitih hardverskih realizacija dvodimenzionalnog inverznog DWT 5/3 filtra

Da bi se izvršilo poređenje rezultata sinteze 2-D inverzne DWT 5/3 arhitekture opisane u ovoj disertaciji sa najboljim raspoloživim rezultatima sinteze drugih realizacija iz literature, 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji je implementirana u *Xilinx FPGA* čipovima *Virtex-4 XC4VFX100* i *Virtex-5 XC5VLX110T*.

Rezultati sinteze različitih 2-D inverznih DWT 5/3 arhitektura sa 16-bitnom dužinom reči su prikazani u tabeli 4.4. Može se uočiti da 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji zahteva korišćenje najmanjeg broja konfigurabilnih logičkih blokova (*CLB*) u poređenju sa arhitekturom [70] i PMA arhitekturom [73]. Takođe, 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji koristi manji broj *CLB* blokova i od RMA arhitekture [73], iako je realizacija iz ove disertacije implementirana za procesiranje slika dimenzija 512×512 sa pet nivoa kompozicije, dok je RMA arhitektura [73] implementirana za procesiranje slika dimenzija 256×256 sa samo tri nivoa kompozicije. Sa druge strane, s obzirom da PMA arhitektura [73] koristi *pipeline* procesor u sklopu svog dizajna, ona omogućava postizanje najveće radne frekvencije, dok realizacija iz ove disertacije omogućava postizanje veće radne frekvencije od arhitekture [70].

Poređenje utrošenih memorijskih resursa različitih 2-D inverznih DWT 5/3 arhitektura je predstavljeno u tabeli 4.5. U tu svrhu, 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji je implementirana sa 10-bitnom dužinom reči kako bi se omogućilo pravilno poređenje sa rezultatima objavljenim u [70]. Rezultati implementacije jasno pokazuju da 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji zahteva korišćenje memorije najmanjeg kapaciteta u poređenju sa ostalim arhitekturama, i pored toga što je arhitektura iz ove disertacije implementirana za procesiranje slika dimenzija 512×512 sa pet nivoa kompozicije, dok su arhitekture [49] i [88-90] implementirane za procesiranje slika sa samo jednim nivoom kompozicije i u nekim slučajevima za procesiranje slika manjih dimenzija.

Tabela 4.4. Rezultati sinteze različitih 2-D inverznih DWT 5/3 arhitektura

2-D inverzna DWT 5/3 arhitektura	[70]	[73] PMA	Realizacija iz ove disertacije	[73] RMA	Realizacija iz ove disertacije
Broj nivoa kompozicije	5	5	5	3	5
Dimenzije slike	512×512	512×512	512×512	256×256	512×512
FPGA čip	XC5VLX110T	XC5VLX110T	XC5VLX110T	XC4VFX100	XC4VFX100
Broj CLB blokova	1052	1178	1049	1040	894
Maksimalna radna frekvencija [MHz]	221	539	248	n/a	164

Tabela 4.5. Poređenje utrošenih memorijskih resursa različitih 2-D inverznih DWT 5/3 arhitektura

2-D inverzna DWT 5/3 arhitektura	[49]	[88]	[89]	[90]	[70]	Realizacija iz ove disertacije
Broj nivoa kompozicije	1	1	1	1	5	5
Dimenzije slike	128×128	256×256	512×512	512×512	512×512	512×512
Utrošena memorija [kbit]	236	>57	72	128	32	20

Tabela 4.6. Disipacija različitih 2-D inverznih DWT 5/3 arhitektura na 100MHz (za *Virtex-5 XC5VLX110T FPGA* čip)

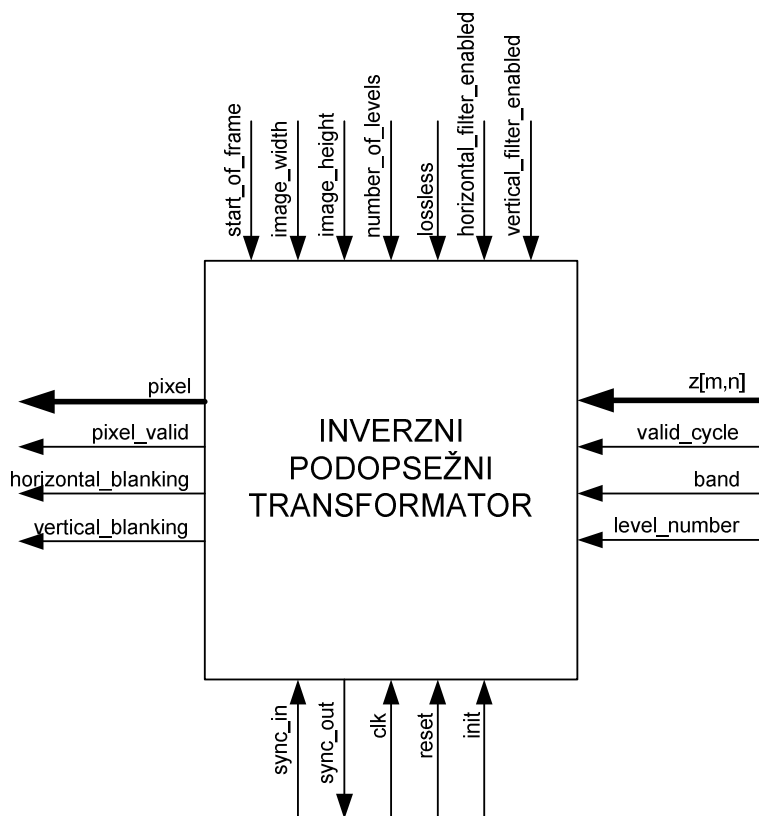
2-D inverzna DWT 5/3 arhitektura	[70]	[73] PMA	Realizacija iz ove disertacije
Broj nivoa kompozicije	1	1	5
Ukupna disipacija [mW]	1221	1009	1223

U tabeli 4.6 su predstavljeni rezultati analize disipacije na radnoj frekvenciji od 100MHz za slike dimenzija 512×512, za implementacije sa 16-bitnom dužinom reči u FPGA čipu *Virtex-5 XC5VLX110T*. Može se uočiti da 2-D inverzna DWT 5/3 arhitektura opisana u ovoj disertaciji ima sličnu disipaciju kao i arhitektura [70] i PMA arhitektura [73], iako je disipacija za arhitekturu [70] i PMA arhitekturu [73] procenjena

za realizacije sa samo jednim nivoom kompozicije, dok je disipacija za arhitekturu opisano u ovoj disertaciji procenjena za realizaciju sa pet nivoa kompozicije.

4.4. Blok šema najvišeg hijerarhijskog nivoa inverznog podopsežnog transformatora

Na slici 4.20 je prikazan najviši hijerarhijski nivo hardverske realizacije inverznog podopsežnog transformatora.



Slika 4.20. Najviši hijerarhijski nivo hardverske realizacije inverznog podopsežnog transformatora

Ulazni podaci za hardversku realizaciju inverznog podopsežnog transformatora su:

- $z[m,n]$: podaci koji predstavljaju komponente dekomponovanog signala;

Kontrolni signali i podaci koji se dovode na ulaz inverznog podopsežnog transformatora su:

- *init*: kontrolni signal kojim se pokreće postupak inicijalizacije dekodera (a time i inverznog podopsežnog transformatora);
- *reset*: kontrolni signal za resetovanje dekodera (a time i inverznog podopsežnog transformatora);
- *clk*: signal takta;
- *sync_in*: ulazni kontrolni signal za sinhronizaciju dekodera za različite ravni boje;
- *start_of_frame*: kontrolni signal koji označava početni podatak na nivou frejma za svaki nivo kompozicije ponaosob;
- *image_width*: kontrolni parametar koji definiše širinu frejma video sekvence koja se dekomprimuje;
- *image_height*: kontrolni parametar koji definiše visinu frejma video sekvence koja se dekomprimuje;
- *number_of_levels*: kontrolni parametar koji označava ukupan broj nivoa kompozicije;
- *lossless*: kontrolni parametar koji određuje da li se dekompresija video sekvence vrši sa ili bez gubitaka;
- *horizontal_filter_enabled*: kontrolni parametar koji određuje da li su i na kojim nivoima kompozicije inverzni horizontalni filtri unutar inverznog podopsežnog transformatora uključeni;
- *vertical_filter_enabled*: kontrolni parametar koji određuje da li su i na kojim nivoima kompozicije inverzni vertikalni filtri unutar inverznog podopsežnog transformatora uključeni;
- *valid_cycle*: kontrolni signal koji kvalifikuje ulazne podatke $z[m,n]$ kao validne ili nevalidne u tekućem ciklusu takta;
- *band*: podatak koji označava podopseg kojem pripada ulazni podatak $z[m,n]$;
- *level_number*: podatak koji označava nivo kompozicije kojem pripada ulazni podatak $z[m,n]$.

Izlazni podatak inverznog podopsežnog transformatora je:

- *pixel*: podaci koji predstavljaju dekomprimovanu video sekvencu; reč je zapravo o podacima $w[m,n]$ sa slike 4.12.

Osim navedenog izlaznog podatka, postoji još i nekoliko kontrolnih signala:

- *pixel_valid*: kontrolni signal koji ukazuje na validnost izlaznog podatka *pixel* u tekućem ciklusu takta;
- *horizontal_blanking*: kontrolni signal koji definiše trajanje pauze između susednih linija frejma dekomprimovane video sekvence;
- *vertical_blanking*: kontrolni signal koji definiše trajanje pauze između susednih frejmova dekomprimovane video sekvence;
- *sync_out*: izlazni kontrolni signal za sinhronizaciju dekodera za različite ravni boje.

5. Realizacija brzog dekodera za jednu ravan boje

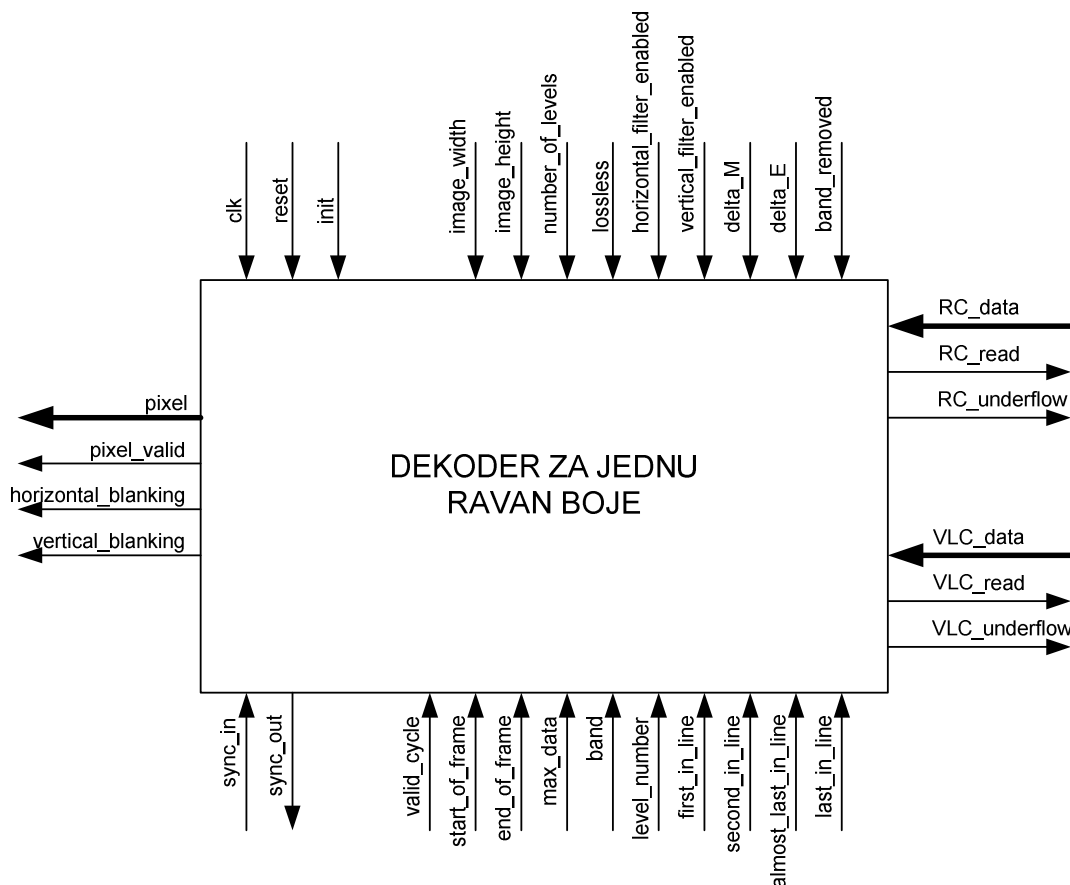
Na slici 5.1 je prikazan najviši hijerarhijski nivo brzog dekodera komprimovane slike za jednu ravan boje. Ulazni podaci dekodera za jednu ravan boje su:

- *RC_data*: podaci koji predstavljaju komprimovanu video sekvencu, kodovani pomoću enkodera opsega; reč je o podacima koji se na ulaz dekodera za jednu ravan boje dovode za svaki nivo kompozicije ponaosob;
- *VLC_data*: podaci koji predstavljaju komprimovanu video sekvencu, kodovani pomoću VLC enkodera (enkodera sa promenljivom dužinom koda); reč je o podacima koji se na ulaz dekodera za jednu ravan boje dovode za svaki nivo kompozicije ponaosob.

Kontrolni signali i podaci koje generiše blok "SEKVENCER" (koji je opisan u narednom poglavlju), a koji se dovode na ulaz dekodera za jednu ravan boje su:

- *valid_cycle*: kontrolni signal koji kvalifikuje ulazne podatke kao validne ili nevalidne;
- *start_of_frame*: kontrolni signal koji označava početni podatak na nivou frejma za svaki nivo ponaosob;
- *end_of_frame*: kontrolni signal koji označava zadnji podatak na nivou frejma za sve nivoe zajedno;
- *max_data*: kontrolni signal koji označava da li se u trenutnom ciklusu dekoduje maksimum na sledeće četiri linije podataka za svaki nivo i za svaki podopseg ponaosob (koji potom koristi VLC dekodera);
- *band*: podatak koji označava podopseg u okviru tekućeg nivoa kompozicije kojem pripada ulazni podatak;
- *level_number*: podatak koji označava nivo kompozicije kojem pripada ulazni podatak;

- *first_in_line*: kontrolni signal koji označava prvi podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob;
- *second_in_line*: kontrolni signal koji označava drugi podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob;
- *almost_last_in_line*: kontrolni signal koji označava predzadnji podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob;
- *last_in_line*: kontrolni signal koji označava zadnji podatak u tekućoj liniji za svaki od podopsega i za svaki nivo ponaosob.



Slika 5.1. Najviši hijerarhijski nivo brzog dekodera komprimovane slike za jednu ravan boje

Osim kontrolnih signala i podataka koje generiše blok “SEKVENCER”, i sledeći kontrolni signali i podaci se dovode na ulaz dekodera za jednu ravan boje:

- *init*: kontrolni signal kojim se pokreće postupak inicijalizacije dekodera;

- *reset*: kontrolni signal za resetovanje dekodera;
- *clk*: signal takta;
- *sync_in*: ulazni kontrolni signal za sinhronizaciju dekodera za različite ravni boje;
- *image_width*: kontrolni parametar koji definiše širinu frejma video sekvence koja se dekomprimuje;
- *image_height*: kontrolni parametar koji definiše visinu frejma video sekvence koja se dekomprimuje;
- *number_of_levels*: kontrolni parametar koji označava ukupan broj nivoa kompozicije;
- *lossless*: kontrolni parametar koji određuje da li se dekompresija video sekvence vrši sa ili bez gubitaka;
- *horizontal_filter_enabled*: kontrolni parametar koji određuje da li su i na kojim nivoima kompozicije inverzni horizontalni filtri unutar inverznog podopsežnog transformatora uključeni;
- *vertical_filter_enabled*: kontrolni parametar koji određuje da li su i na kojim nivoima kompozicije inverzni vertikalni filtri unutar inverznog podopsežnog transformatora uključeni;
- *delta_M*: kontrolni parametar koji definiše mantisu za osnovni korak dekvantizacije;
- *delta_E*: kontrolni parametar koji definiše eksponent za osnovni korak dekvantizacije;
- *band_removed*: kontrolni parametar koji određuje da li su i koji podopsezi uklonjeni iz procesa dekompresije.

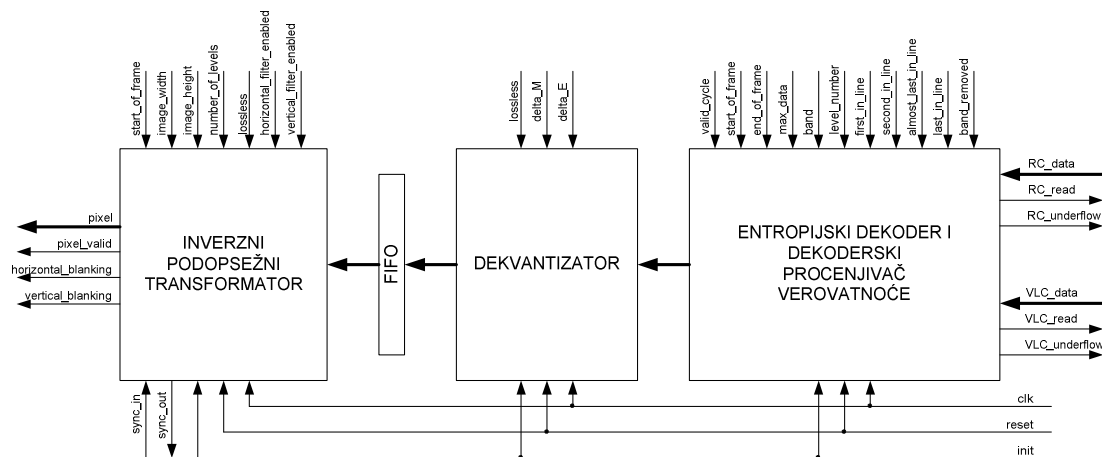
Izlazni podatak dekodera za jednu ravan boje je:

- *pixel*: podaci koji predstavljaju dekomprimovanu video sekvencu.

Osim navedenog izlaznog podatka, postoji još i niz izlaznih kontrolnih signala:

- *pixel_valid*: kontrolni signal koji ukazuje na validnost izlaznog podatka *pixel* u tekućem ciklusu takta;

- *horizontal_blanking*: kontrolni signal koji definiše trajanje pauze između susednih linija frejma dekomprimovane video sekvence;
- *vertical_blanking*: kontrolni signal koji definiše trajanje pauze između susednih frejmova dekomprimovane video sekvence;
- *sync_out*: izlazni kontrolni signal za sinhronizaciju dekodera za različite ravni boje;
- *RC_underflow*: kontrolni signal (za svaki nivo kompozicije ponaosob) koji ukazuje da je narušena dinamika procesa dekompresije usled neblagovremenog učitavanja potrebnih podataka preko ulaza *RC_data*;
- *RC_read*: kontrolni signal (za svaki nivo kompozicije ponaosob) kojim se kontroliše učitavanje potrebnih podataka preko ulaza *RC_data*;
- *VLC_underflow*: kontrolni signal (za svaki nivo kompozicije ponaosob) koji ukazuje da je narušena dinamika procesa dekompresije usled neblagovremenog učitavanja potrebnih podataka preko ulaza *VLC_data*;
- *VLC_read*: kontrolni signal (za svaki nivo kompozicije ponaosob) kojim se kontroliše učitavanje potrebnih podataka preko ulaza *VLC_data*.



Slika 5.2. Interna struktura brzog dekodera komprimovane slike za jednu ravan boje

Interna struktura brzog dekodera komprimovane slike za jednu ravan boje je prikazana na slici 5.2. Interna struktura dekodera komprimovane slike za jednu ravan boje se sastoji od tri bloka ("ENTROPIJSKOG DEKODERA I DEKODERSKOG

PROCENJIVAČA VEROVATNOĆE”, “DEKVANTIZATORA” i “INVERZNOG PODOPSEŽNOG TRANSFORMATORA”), čije su realizacije već opisane u prethodnim poglavljima ove disertacije. Na izlazu bloka “ENTROPIJSKI DEKODER I DEKODERSKI PROCENJIVAČ VEROVATNOĆE” se generišu odbirci signala, koji se potom u bloku “DEKVANTIZATOR” dekvantizuju, u slučaju kompresije/dekompresije sa gubicima, ili samo prosleđuju bloku “INVERZNI PODOPSEŽNI TRANSFORMATOR” u slučaju kompresije/dekompresije bez gubitaka. Pre nego što “INVERZNI PODOPSEŽNI TRANSFORMATOR” prihvati odbirke signala koji predstavljaju izlaz “DEKVANTIZATORA”, oni se smeštaju u odgovarajuću FIFO memoriju, kako bi se dinamika kojom se na izlazu “DEKVANTIZATORA” pojavljuju odbirci signala uskladila sa dinamikom prijema tih odbiraka signala od strane “INVERZNOG PODOPSEŽNOG TRANSFORMATORA”.

Simulacija i funkcionalna verifikacija opisane realizacije brzog dekodera komprimovane slike za jednu ravan boje je izvršena korišćenjem softverskog alata *Modelsim SE 6.5d*. Rezultati simulacije su upoređeni sa softverski generisanim teorijski očekivanim rezultatima, i u oba slučaja su dobijeni identični rezultati, čime je potvrđena funkcionalna ispravnost realizovane strukture.

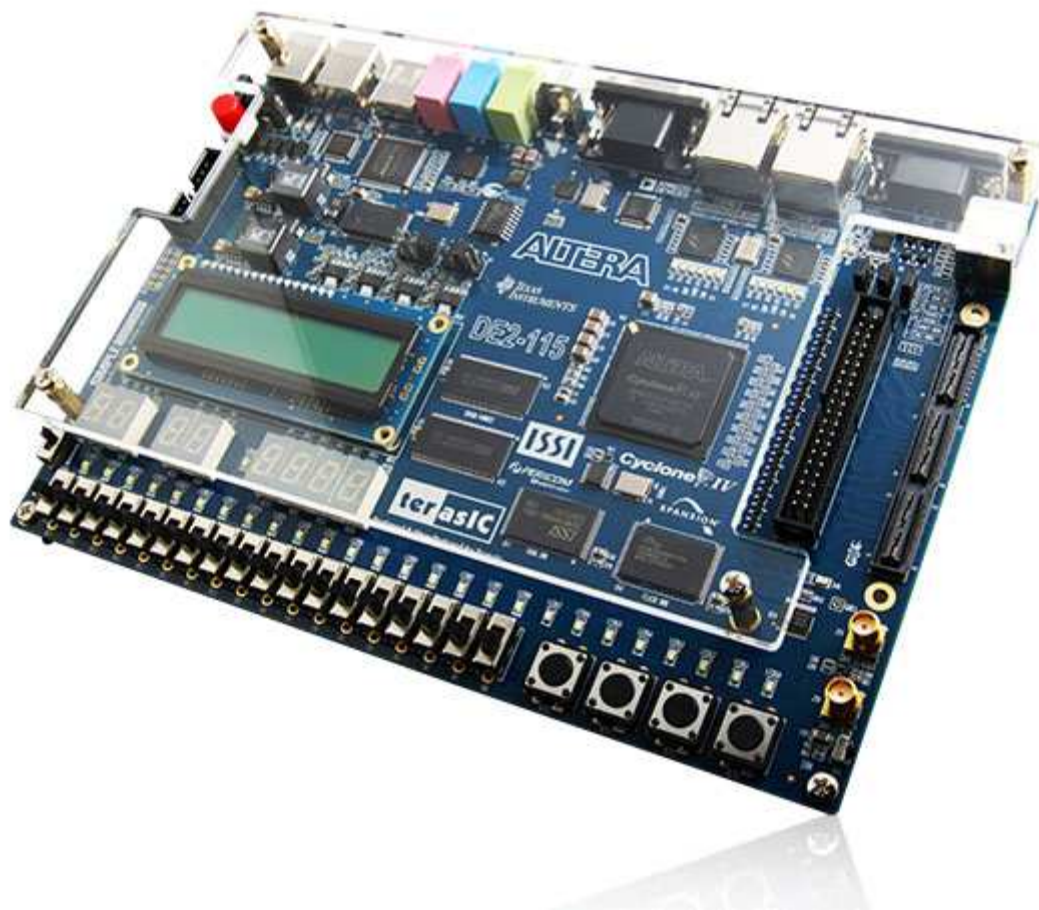
Sinteza hardverske realizacije brzog dekodera komprimovane slike za jednu ravan boje, koja podržava dekompresiju sekvence HD rezolucije (1920x1080 piksela) sa sedam nivoa kompozicije, izvršena je korišćenjem softverskog paketa *Altera Quartus II 10.0* za *FPGA* čip EP4CE115F29C7. Rezultati sinteze su prezentovani u tabeli 5.1.

Tabela 5.1. Hardverski resursi i performanse hardverske realizacije brzog dekodera komprimovane slike za jednu ravan boje u čipu *Altera EP4CE115F29C7*

Broj logičkih elemenata	28641
Kapacitet korišćene memorije	627784 bita
Broj množača (9-bitnih)	4
Maksimalna radna frekvencija dizajna na 85°C	125,26 MHz

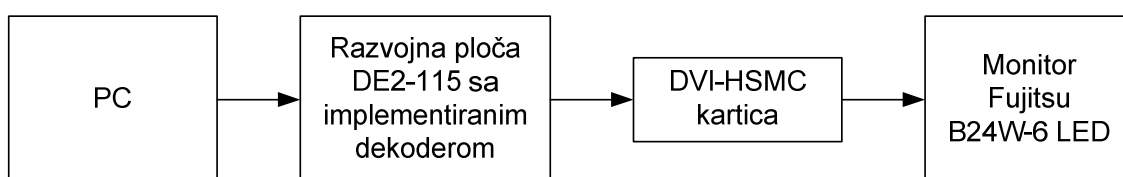
6. Realizacija i implementacija brzog dekodera sa minimalnim utrošenim resursima za sliku u boji

Dekoder koji je opisan u ovoj disertaciji je implementiran na razvojnoj ploči *Altera DE2-115* proizvođača *Terasic Technologies* [91], koja je prikazana na slici 6.1.



Slika 6.1. Razvojna ploča *Altera DE2-115*

Na ovoj razvojnoj ploči se, između ostalih, nalazi i komponenta (*FPGA* čip) EP4CE115F29C7 iz familije *Cyclone IVE* proizvođača *Altera*, koja raspolaže sa 114480 osnovnih logičkih ćelija, 3888 kbita ugrađene memorije koja je organizovana u vidu 432 M9K bloka, 4 fazno kontrolisane petlje, 532 ugrađena 9-bitna brojača i 529 pinova [92]. Na razvojnoj ploči postoji i nekoliko tipova spoljašnjih memorija, i to: *SDRAM* kapaciteta 128MB, *SRAM* kapaciteta 2MB, fleš memorija kapaciteta 8MB i *EEPROM* kapaciteta 32kbita. Ploča sadrži i dva mrežna interfejsa.



Slika 6.2. Blok dijagram eksperimentalne postavke korišćene za verifikaciju brzog dekodera komprimovane slike



Slika 6.3. DVI-HSMC kartica

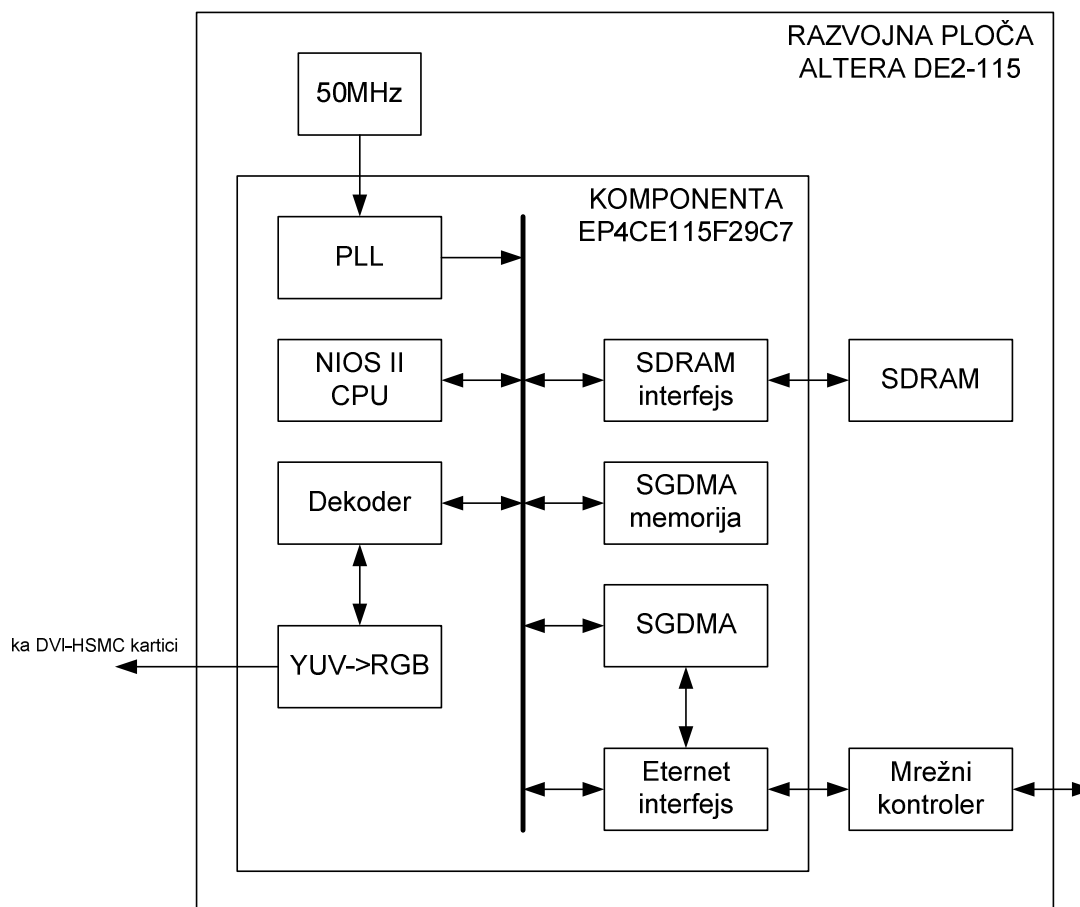
Za potrebe verifikacije brzog dekodera komprimovane slike, opisanog u ovoj disertaciji, formirana je eksperimentalna postavka, čiji je blok dijagram prikazan na slici 6.2. Personalni računar je preko mrežnog kabla povezan sa razvojnom pločom DE2-

115, u čijem *FPGA* čipu je implementiran brzi dekomprimovane slike. Mrežnim kablom se prosleđuju komande dekomprimovane slike, uključujući i komande za setovanje parametara dekomprimovane slike, kao i komanda za start procesa dekomprimovane slike. Preko mrežnog kabla se dekomprimovane slike i podaci koji predstavljaju komprimovanu sliku, čiju dekomprimaciju treba da obavi dekomprimovane slike. Mrežni server je softverski implementiran u procesoru *NIOS II* koji je jedna od komponenti unutar *FPGA* čipa. U komunikaciji između personalnog računara i razvojne ploče se koriste *TCP* i *UDP* mrežni paketi, koji služe za kontrolu dekomprimovane slike i prenos komprimovane slike, respektivno. S obzirom da se dekomprimovana slika, po postavljenoj eksperimentalnoj postavci, prikazuje na monitoru sa *DVI* ulazom, i s obzirom da razvojna ploča *DE2-115* ne poseduje *DVI* izlaz, neophodno je za povezivanje razvojne ploče sa monitorom iskoristiti odgovarajuću adaptersku karticu. U tu svrhu je upotrebljena *DVI-HSMC* kartica [93], koja je prikazana na slici 6.3. Njena uloga je da pinove sa *HSMC* konektora sa razvojne ploče *DE2-115*, na pravilan način poveže sa odgovarajućim pinovima na svom *DVI* izlazu. *DVI* konektor sa *DVI-HSMC* kartice se dalje preko *DVI* kabla povezuje sa monitorom *Fujitsu B24W-6 LED* [94], koji je prikazan na slici 6.4, i na kojem se prikazuje dekomprimovana slika.



Slika 6.4. *Fujitsu B24W-6 LED* monitor

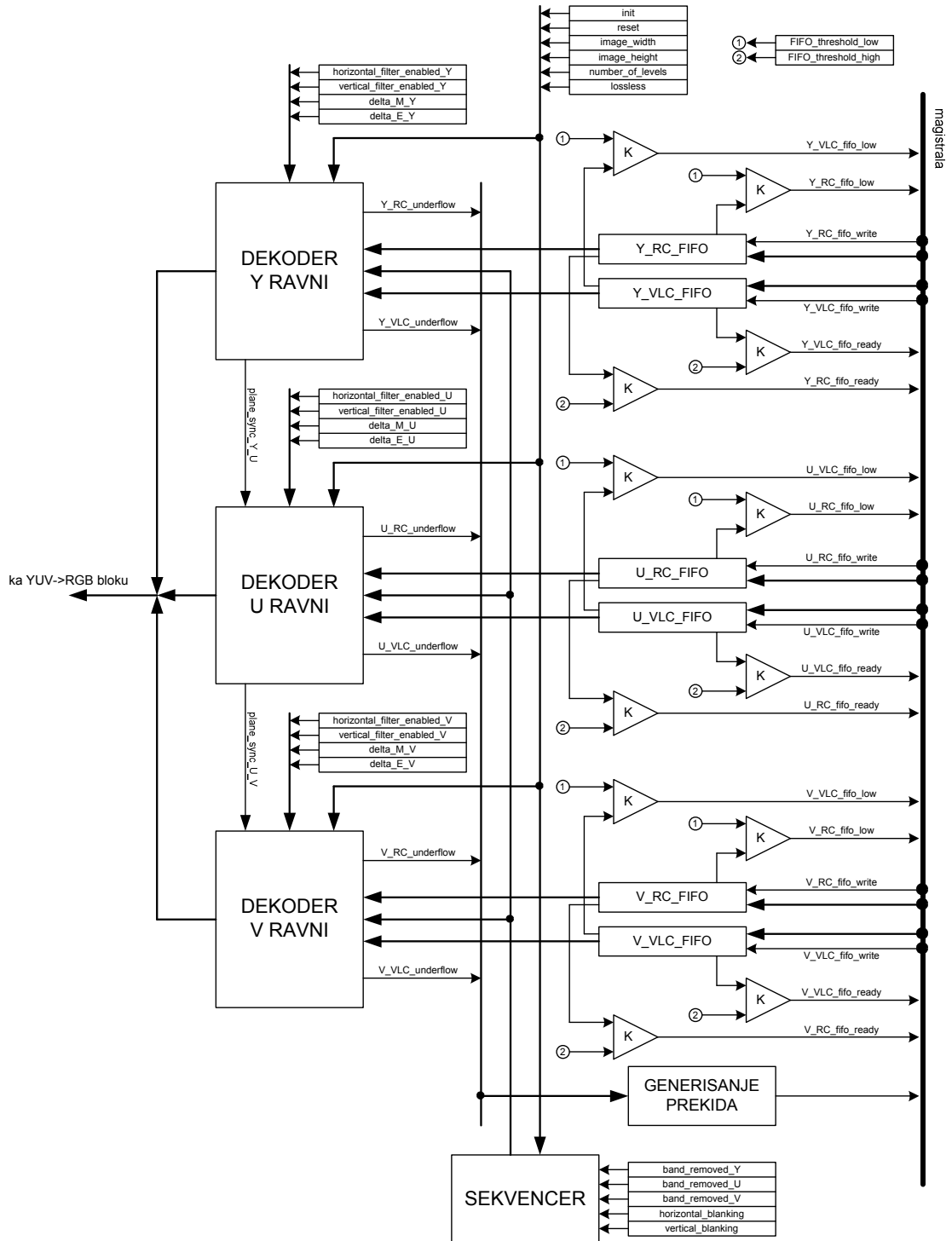
Blok dijagram hardverske realizacije sistema za dekomprimaciju digitalne slike u boji, koji sadrži brzi dekomprimovane slike, prikazan je na slici 6.5.



Slika 6.5. Blok dijagram hardverske realizacije sistema za dekompresiju digitalne slike u boji

Signali takta za sve komponente u sistemu se generišu pomoću oscilatora koji radi na frekvenciji od 50MHz i fazno kontrolisane petlje (*PLL*). Komunikacija između *NIOS II* procesora i ostalih komponenti sistema se vrši preko 32-bitne Avalon magistrale [95]. Kao programska memorija i memorija podataka za procesor, koristi se *SDRAM* na razvojnoj ploči *DE2-115*. *SGDMA* kontroler ima ulogu da obezbedi brzi prenos podataka između *SDRAM*-a i mrežnog kontrolera koji se takođe nalazi na razvojnoj ploči. Da bi *SGDMA* kontroler ispravno funkcionisao, neophodno je da se čuva lista njegovih poslova, i u tu svrhu se koristi *SGDMA* memorija, implementirana u okviru *FPGA* čipa. Unutar *FPGA* čipa je implementirana i komponenta koja predstavlja brzi dekoder komprimovane slike opisan u ovoj disertaciji. Izlazi dekodera su pikseli dekomprimovane slike u *YUV* formatu, koji se potom pomoću odgovarajuće

komponente, takođe implementirane u *FPGA* čipu, konvertuju u RGB format, kako bi dalje bili prosleđeni ka *DVI-HSMC* kartici.



Slika 6.6. Principska blok šema komponente koja predstavlja brzi dekoder komprimovane slike opisan u ovoj disertaciji

Principijska blok šema komponente koja predstavlja brzi dekomprimovane slike opisan u ovoj disertaciji, predstavljena je na slici 6.6. Interfejs prema magistrali nije prikazan, kako bi prikaz blok šeme bio pregledniji. Dekoder komprimovane slike sadrži set kontrolnih registara, čiji sadržaj upisuje procesor preko magistrale, a koji su zaduženi za kontrolu rada dekodera i setovanje parametara dekodera. Neki od kontrolnih registara sadrže kontrolne parametre koji se odnose na dekomprimovane slike u celini (tj. na sve tri ravni boje), i to su:

- *init*: kontrolni parametar kojim se pokreće postupak inicijalizacije dekodera;
- *reset*: kontrolni signal za resetovanje dekodera;
- *width*: kontrolni parametar koji definiše širinu frejma video sekvence koja se dekomprimuje;
- *height*: kontrolni parametar koji definiše visinu frejma video sekvence koja se dekomprimuje;
- *number_of_levels*: kontrolni parametar koji definiše ukupan broj nivoa kompozicije;
- *lossless*: kontrolni parametar koji određuje da li se dekompresija video sekvence vrši sa ili bez gubitaka;
- *horizontal_blanking*: kontrolni parametar koji definiše trajanje pauze između susednih linija frejma video sekvence koja se dekomprimuje;
- *vertical_blanking*: kontrolni parametar koji definiše trajanje pauze između susednih frejmova video sekvence koja se dekomprimuje;
- *FIFO_threshold_low*: kontrolni parametar koji definiše donji prag u vidu broja podataka koji se nalaze u FIFO memorijama i na osnovu koga se signalizira da je odgovarajuća FIFO memorija skoro prazna i da je neophodno izvršiti upis dodatnih podataka u tu FIFO memoriju;
- *FIFO_threshold_high*: kontrolni parametar koji definiše prag u vidu broja podataka koji se nalaze u FIFO memorijama i na osnovu koga se signalizira da se u odgovarajuću FIFO memoriju može izvršiti upis dodatnih podataka iako ona nije skoro prazna.

Ostali kontrolni registri sadrže kontrolne parametre koji se odnose na pojedinačne ravni boje unutar dekodera, pri čemu postoje tri skupa takvih registara (za Y, U ili V ravan boje) i to su:

- *horizontal_filter_enabled*: kontrolni parametar koji određuje da li su i na kojim nivoima kompozicije inverzni horizontalni filtri unutar inverznog podopsežnog transformatora uključeni;
- *vertical_filter_enabled*: kontrolni parametar koji određuje da li su i na kojim nivoima kompozicije inverzni vertikalni filtri unutar inverznog podopsežnog transformatora uključeni;
- *delta_M*: kontrolni parametar koji definiše mantisu za osnovni korak dekvantizacije;
- *delta_E*: kontrolni parametar koji definiše eksponent za osnovni korak dekvantizacije;
- *band_removed*: kontrolni parametar koji određuje da li su i koji podopsezi uklonjeni iz procesa dekompresije.

Nakon početka procesa dekompresije, koji se inicira od strane personalnog računara, *NIOS II* procesor upisuje bajt po bajt podatke koje dobija od personalnog računara, a koji predstavljaju komprimovanu sliku, u sledeće FIFO memorije: “Y_RC_FIFO”, “Y_VLC_FIFO”, “U_RC_FIFO”, “U_VLC_FIFO”, “V_RC_FIFO” i “V_VLC_FIFO”. Imajući u vidu da je u sklopu ove doktorske disertacije implementiran dekodер koji podržava dekompresiju sa sedam nivoa kompozicije, svaki od ovih blokova koji su označeni na slici 6.6 kao FIFO memorije, zapravo sadrži 7 nezavisnih FIFO memorija i to za svaki nivo kompozicije po jednu. Ukoliko je broj podataka u bilo kojoj od ovih FIFO memorija manji od parametra *FIFO_threshold_low* (što se utvrđuje odgovarajućim poređenjima koje vrše komparatori “K”), aktivira se odgovarajući signal *Y_RC_fifo_low*, *Y_VLC_fifo_low*, *U_RC_fifo_low*, *U_VLC_fifo_low*, *V_RC_fifo_low* ili *V_VLC_fifo_low* (reč je o zasebnim signalima za svaki od nivoa kompozicije ponaosob), koji procesoru signalizira da je odgovarajuća FIFO memorija skoro prazna i da je neophodno izvršiti upis dodatnih podataka u tu FIFO memoriju kako pravilan dotok podataka dekodерima pojedinih ravni boja u skladu sa zadatom dinamikom ne bi bio ugrožen. Ukoliko je broj podataka u bilo kojoj od ovih FIFO memorija manji od

parametra *FIFO_threshold_high* (što se utvrđuje odgovarajućim poređenjima koje vrše komparatori “K”), aktivira se odgovarajući signal *Y_RC_fifo_ready*, *Y_VLC_fifo_ready*, *U_RC_fifo_ready*, *U_VLC_fifo_ready*, *V_RC_fifo_ready* ili *V_VLC_fifo_ready* (reč je, takođe, o zasebnim signalima za svaki od nivoa kompozicije ponaosob), koji procesoru signalizira da je u odgovarajuću FIFO memoriju moguće izvršiti upis dodatnih podataka iako ona nije skoro prazna (tj. iako nije ugrožen pravilan dotok podataka dekoderima pojedinih ravni boja u skladu sa zadatom dinamikom).

Podatke iz pomenutih FIFO memorija potom preuzimaju blokovi “DEKODER Y RAVNI”, “DEKODER U RAVNI” i “DEKODER V RAVNI”, koji vrše proces dekompresije slike na nivou pojedinačnih ravni boja i to u skladu sa dinamikom koju definiše blok “SEKVENCER”. Sinhronizacija rada ova tri bloka dekodera za svaku od ravni boja se vrši pomoću signala *plane_sync_Y_U* i *plane_sync_U_V*. Izlazni podaci blokova dekodera za pojedinačne ravni boja su 8-bitne Y, U i V komponente piksela dekomprimovane slike. Blokovi “DEKODER Y RAVNI”, “DEKODER U RAVNI” i “DEKODER V RAVNI” generišu signale *Y_RC_underflow*, *Y_VLC_underflow*, *U_RC_underflow*, *U_VLC_underflow*, *V_RC_underflow* i *V_VLC_underflow*, respektivno, i to za svaki od 7 nivoa kompozicije ponaosob, u slučaju se desi da blokovi dekodera (tačnije “VLC DEKODER” i/ili “DEKODER OPSEGA”) imaju potrebu da očitaju i procesiraju podatak iz sopstvenih prihvatnih registara i/ili memorija (u koje se smeštaju podaci dobijeni od FIFO memorija sa slike 6.6), a da pri tome traženi podatak ne postoji (tj. da nije na vreme odgovarajući podatak upisan u prihvatne registre i/ili memoriju). U tom slučaju se pomenuti signali prosleđuju bloku “GENERISANJE PREKIDA” koji potom generiše prekidni zahtev procesoru, koji ima za cilj prekidanje postupka dekompresije slike jer je potrebna dinamika za proces dekompresije narušena.

Sinteza hardverske realizacije brzog dekodera komprimovane slike, opisanog u ovoj disertaciji, čija je principna blok šema prikazana na slici 6.6, izvršena je korišćenjem softverskog paketa *Altera Quartus II 10.0* za *FPGA* čip *EP4CE115F29C7*. Rezultati sinteze su prezentovani u tabeli 6.1.

Prezentovani rezultati sinteze se odnose na dekodera koji omogućava proces dekompresije slike sa sledećim parametrima:

- rezolucija slike: HD rezolucija (1920x1080 piksela);
- brzina video sekvence: do 55 slika u sekundi;
- format dekomprimovane slike 4:2:0.

Tabela 6.1. Hardverski resursi i performanse hardverske realizacije brzog dekodera komprimovane slike u čipu *Altera EP4CE115F29C7*

Broj logičkih elemenata	77127
Kapacitet korišćene memorije	1884207 bita
Broj množača (9-bitnih)	12
Maksimalna radna frekvencija dizajna na 85°C	114,71 MHz

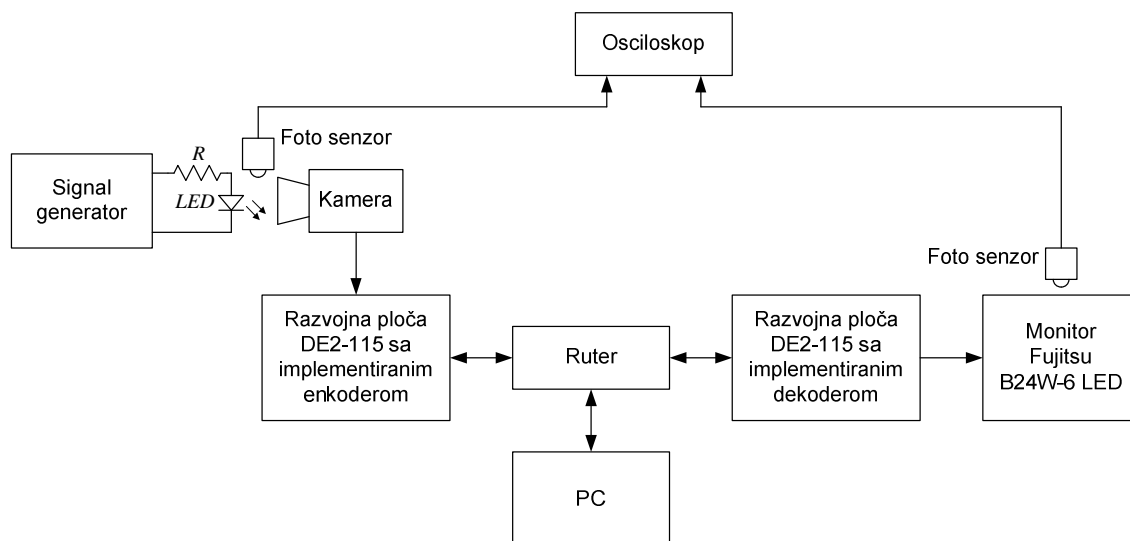
Rezultati sinteze hardverske realizacije sistema za dekompresiju digitalne slike u boji, čiji je blok dijagram prikazan na slici 6.5, izvršene korišćenjem softverskog paketa *Altera Quartus II 10.0* za *FPGA* čip *EP4CE115F29C7*, su prezentovani u tabeli 6.2.

Tabela 6.2. Hardverski resursi i performanse sistema za dekompresiju slike sa mrežnim interfejsom u čipu *Altera EP4CE115F29C7*

Broj logičkih elemenata	88522
Kapacitet korišćene memorije	2610793 bita
Broj množača (9-bitnih)	16
Maksimalna radna frekvencija dizajna na 85°C	111,34 MHz

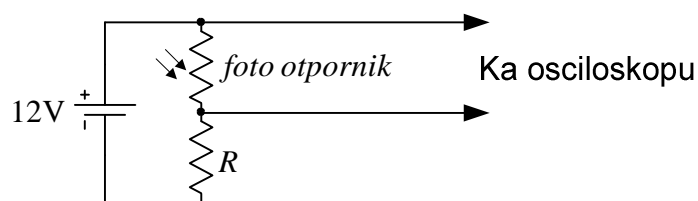
Na slici 6.7 je prikazana eksperimentalna postavka pomoću koje je izvršeno merenje kašnjenja koje proizvodi sistem za kompresiju i dekompresiju digitalne slike, koga čine već razvijeni hardverski enkoder [30] i brzi hardverski dekodeer opisan u ovoj disertaciji. Signal generator pobuđuje svetleću diodu (*LED*), koja se naizmenično pali i gasi sa frekvencijom 1Hz. Naizmenično uključivanje i isključivanje *LED* se snima kamerom koja je povezana na razvojnu ploču *DE2-115*, na kojoj je implementiran enkoder za kompresiju digitalne slike [30]. Ovaj enkoder vrši kompresiju video sekvence koju snima kamera i komprimovanu sekvencu prosleđuje preko rutera dekodeeru opisanom u ovoj disertaciji, koji je implementiran na drugoj razvojnoj ploči

DE2-115. Dekoder vrši dekompresiju komprimovane video sekvence, nakon čega je prikazuje na monitoru koji je povezan sa dekodrom. Jedan foto senzor registruje paljenje i gašenje *LED* pobuđene signal generatorom, dok drugi foto senzor registruje sliku tog događaja na monitoru, koja se prikazuje sa zakašnjenjem koje unosi sistem koga čine enkoder, ruter i dekodeer. Signali sa oba foto senzora se mere osciloskopom, kao i njihov međusobni fazni pomak. Na taj način se meri kašnjenje koje unosi sistem koga čine enkoder, ruter i dekodeer. Personalni računar na početku celog procesa preko rutera setuje mrežne parametre enkodera i dekodera, kako bi proces prenosa komprimovane sekvence preko mreže, između enkodera i dekodera bio funkcionalan. Takođe, preko personalnog računara se i startuje proces kompresije, prenosa i dekompresije video sekvence koju snima kamera u realnom vremenu.



Slika 6.7. Blok dijagram eksperimentalne postavke korišćene za merenje kašnjenja koje proizvodi sistem za kompresiju i dekompresiju digitalne slike

Na slici 6.8 je prikazano na koji način su realizovani foto senzori koji se koriste u eksperimentalnoj postavci sa slike 6.7.



Slika 6.8. Implementacija foto senzora korišćenog u postavci sa slike 6.7.

U tabeli 6.3 je predstavljeno kašnjenje koje je određeno opisanim eksperimentom. Ovim eksperimentom je takođe i dodatno potvrđena funkcionalna ispravnost brzog dekodera komprimovane slike koji je opisan u ovoj disertaciji, kao i njegova kompatibilnost sa enkoderom [30].

Tabela 6.3. Kašnjenje koje proizvodi sistem za kompresiju i dekompresiju digitalne slike

Sistem	Kašnjenje
Enkoder [30] + ruter + dekodeer	160ms

Imajući u vidu da proces dekompresije digitalne slike prema realizaciji opisanoj u ovoj disertaciji započinje tek nakon što se kompletira proces kompresije jednog celog frejma ulazne slike (iz razloga što dekompresija slike započinje procesiranjem podataka koji predstavljaju rezultat kompresije najdubljeg nivoa dekompozicije, koji se generiše na samom kraju postupka kompresije, tj. nakon kompresije svih ostalih nivoa dekompozicije), i da se prvi rezultati dekompresije, tj. pikseli dekomprimovane slike generišu svega nekoliko linija nakon toga, može se zaključiti (imajući u vidu brzinu prikaza dekomprimovane sekvence) da je u sistemu sa slike 6.7 kašnjenje koje u sistem unose hardverski enkoder i dekodeer značajno manje od kašnjenja koje unosi mrežni ruter koji je zadužen za prenos komprimovane sekvence, kao i kašnjenja koje unosi monitor, što otvara mogućnost dodatnog redukovanja kašnjenja u sistemu.

7. Zaključak

U ovoj doktorskoj disertaciji je opisana nova hardverska realizacija dekodera komprimovane slike sa minimalnim zahtevanim resursima. Realizovani dekodер se sastoji od entropijskog dekodera, dekoderskog procenjivača verovatnoće, dekvantizatora i inverznog podopsežnog transformatora. Svaki od navedenih blokova je zasebno realizovan sa ciljem postizanja što veće uštede u utrošku logičkih i memorijskih resursa, ali je istovremeno vođeno računa i o optimizaciji utroška resursa i na nivou dekodera kao celine. Ova doktorska disertacija sadrži sledeće naučne doprinose:

- Predložena je nova, hardverski efikasna, realizacija entropijskog dekodera, koja omogućava smanjenje utrošenih logičkih resursa u odnosu na realizacije entropijskog dekodera prema postojećem stanju tehnike.
- Predložena je nova, hardverski efikasna, realizacija dekoderskog procenjivača verovatnoće, sa optimizovanim utroškom memorijskih i logičkih resursa.
- Predložena je nova realizacija inverznog podopsežnog transformatora koji vrši 2-D inverznu 5/3 diskretnu *wavelet* transformaciju koristeći 20% manje memorijskih resursa od najbolje do sada objavljene realizacije u literaturi.
- U pogledu utroška logičkih resursa, realizovani inverzni podopsežni transformator je optimalniji od bilo koje druge do sada objavljene realizacije u literaturi.
- Kao osnovni gradivni blok realizovane 2-D inverzne 5/3 DWT arhitekture, korišćena je vremenski nestacionarna hardverska realizacija 1-D inverznog DWT 5/3 filtra koja koristi najmanji broj logičkih elemenata i najmanji broj registara, i pri tome ima najmanju ukupnu disipaciju omogućavajući istovremeno rad na najvećoj radnoj frekvenciji u poređenju sa bilo kojom drugom realizacijom 1-D inverznog DWT 5/3 filtra poznatom iz literature.

Pravci daljih istraživanja će biti orijentisani ka dodatnoj optimizaciji i smanjenju utroška logičkih resursa kod 2-D inverzne 5/3 DWT arhitekture, time što bi se umesto

zasebnog horizontalnog 1-D inverznog DWT 5/3 filtra za svaki nivo kompozicije ponaosob, koristio samo jedan horizontalni 1-D inverzni DWT 5/3 filter za sve nivoe. To bi se postiglo odgovarajućim učešljavanjem ciklusa takta u kojima bi različiti nivoi kompozicije pristupali jednom horizontalnom 1-D inverznom DWT 5/3 filtru, na način da u jednom ciklusu takta samo jedan od nivoa kompozicije pristupa i koristi pomenuti horizontalni filter. Slično tome, umesto dva zasebna vertikalna 1-D inverzna DWT 5/3 filtra, koristio bi se samo jedan vertikalni 1-D inverzni DWT 5/3 filter za sve nivoe, što bi se takođe postiglo odgovarajućim učešljavanjem ciklusa takta u kojima bi različiti nivoi kompozicije pristupali jednom vertikalnom 1-D inverznom DWT 5/3 filtru. Osim toga, imajući u vidu široku rasprostranjenost i značaj primene diskretne *wavelet* transformacije, kao i značajne uštede u logičkim i memorijskim resursima koje su ostvarene u realizaciji 2-D inverzne 5/3 DWT arhitekture opisane u ovoj disertaciji, kao logičan pravac daljih istraživanja se nameće i razvoj memorijski i logički efikasne hardverske realizacije trodimenzionalne inverzne 5/3 DWT arhitekture.

Literatura

- [1] S. Todd, G. G. Langdon Jr, and J. Rissanen, "Parameter reduction and context selection for compression of gray-scale images," *IBM J. Res. Develop.*, vol. 29, no. 2, pp. 188-193, Mar. 1985.
- [2] V. R. Algazi and R. R. Estes Jr, "Analysis based coding of image transform and subband coefficients," *SPIE Applications of Digital Image Processing XVIII*, vol. 2564, pp. 11-21, July 1995.
- [3] S. D. Stearns, "Arithmetic coding in lossless waveform compression," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 1874-1879, Aug. 1995.
- [4] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, U.S. Patent 6,222,941, Apr. 2001.
- [5] B. M. Oliver, "Efficient coding," *Bell Syst. Tech. J.*, vol. 31, no. 4, pp. 724-750, July 1952.
- [6] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098-1101, Sep. 1952.
- [7] E. N. Gilbert and E. F. Moore, "Variable length binary encodings," *Bell Syst. Tech. J.*, vol. 38, no. 4, pp. 933-967, July 1959.
- [8] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression", *Commun. ACM*, vol. 30, no. 6, pp. 520-540, June 1987.
- [9] A. Moffat, R. M. Neal, and I. H. Witten., "Arithmetic coding revisited", *Proc. Data Compression Conf.*, Snowbird, UT, pp. 202-211, Mar. 1995.
- [10] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Trans. Inform. Syst.*, vol. 16, no. 3, pp. 256-294, July 1998.
- [11] J. L. Mitchell, and W. B. Pennebaker, "Software implementations of the Q-coder," *IBM J. Res. Develop.*, vol. 21, no. 6, pp. 753-774, Nov. 1988.

- [12] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, and R. B. Arps., "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder," *IBM J. Res. Develop.*, vol. 32, no. 6, pp. 717-726, Nov. 1988.
- [13] W. B. Pennebaker, J. L. Mitchell, U.S. Patent 4,933,883, June 1990.
- [14] W. B. Pennebaker, J. L. Mitchell, U.S. Patent 4,935,882, June 1990.
- [15] L. Bottou, P. G. Howard, and Y. Bengio, "The Z-coder adaptive binary coder," *Proc. Data Compression Conf.*, Snowbird, UT, pp. 13-22, Mar. 1998.
- [16] Y. Bengio, L. Bottou, and P. G. Howard, U.S. Patent 6,188,334, Feb. 2001.
- [17] Y. Bengio, L. Bottou, and P. G. Howard, U.S. Patent 6,225,925, May 2001.
- [18] Y. Bengio, L. Bottou, and P. G. Howard, U.S. Patent 6,281,817, Aug. 2001.
- [19] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18-34, Feb. 1992.
- [20] F. Ono, M. Denki, and K. Kaisha, U.S. Patent 5,059,976, Oct. 1991.
- [21] F. Ono, M. Denki, and K. Kaisha, U.S. Patent 5,307,062, April 1994.
- [22] <http://www.data-compression.info/Algorithms/RC/>
- [23] A. Said and W. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Trans. Image Processing*, vol. 5, no. 9, pp. 1303-1310, Sep. 1996.
- [24] V. Čeperković, S. Pavlović, D. Mirković, and M. Prokin, U.S. patent 8,306,340, Nov. 2012.
- [25] G. N. N. Martin, "Range encoding: an algorithm for removing redundancy from a digitised message," *Proc. Video & Data Recording Conf.*, Southampton, UK, July 24-27, 1979.
- [26] M. Schindler, "A fast renormalization for arithmetic coding," *Poster at DDC, Data Compression Conf.*, Snowbird, UT, Mar. 30 – Apr. 1, 1998.
- [27] <http://www.compressconsult.com/rangecoder/>
- [28] D. J. Magenheimer, L. Peters, K. W. Pettis, and D. Zuras, "Integer multiplication and division on the HP precision architecture," *IEEE Trans. Computers*, vol. 37, no. 8, p. 980-990, Aug. 1988.
- [29] T. Granlund and P. L. Montgomery, "Division by invariant integers using multiplication," *SIGPLAN Notices*, vol. 29, no. 6, p. 61, June 1994.

- [30] V. Rajović, “Hardverska realizacija jednoprolaznog brzog kodeka sa visokim stepenom kompresije i minimalnim zahtevanim resursima,” doktorska disertacija, Univerzitet u Beogradu, Elektrotehnički fakultet, mart 2014.
- [31] K. P. Soman, K. I. Ramachandran, and N. G. Resmi, *Insight into Wavelets from Theory to Practice*, PHI Learning, 2010.
- [32] T. Acharya and P.-S. Tsai, *JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures*, John Wiley and Sons, 2005.
- [33] D. Goldberg, “What every computer scientist should know about floating-point arithmetic,” *ACM Computing Surveys*, vol. 23, no. 1, pp. 5-48, Mar. 1991.
- [34] O. Egger and W. Li, “Subband coding of images using asymmetrical filter banks,” *IEEE Trans. Image Processing*, vol. 4, no. 4, pp. 478-485, Apr. 1995.
- [35] I. Daubechies, “The wavelet transform time-frequency localization and signal analysis,” *IEEE Trans. Inform. Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.
- [36] S. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.
- [37] H. Caglar, Y. Liu, and N. Akansu, “Optimal PR-QMF design for subband image coding,” *J. Visual Commun. Image Representation*, vol. 4, no. 3, pp. 242-253, Sep. 1993.
- [38] O. Egger and W. Li, “Subband coding of images using asymmetrical filter banks,” *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 478-485, Apr. 1995.
- [39] W. Li and O. Egger, “Improved subband coding of images using unequal length PR filters,” *Proc. 14th Gretsri Symp. Signal and Image Process.*, pp. 451-454, Sep. 1993.
- [40] D. Le Gall and A. Tabatabai, “Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques,” *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 761-764, Apr. 1988.
- [41] K. K. Parhi and T. Nishitani, “VLSI architectures for discrete wavelet transforms,” *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 1, no. 2, pp. 191-202, June 1993.

- [42] P. C. Wu and L.G. Chen, "An efficient architecture for two-dimensional discrete wavelet transform," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, no. 4, pp. 536-545, Apr. 2001.
- [43] N. D. Zervas, G. P. Anagnostopoulos, V. Spiliotopoulos, Y. Andreopoulos, and C. E. Goutis, "Evaluation of design alternatives for the 2-D-discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.* vol. 11, no. 12, pp. 1246–1262, Dec. 2001.
- [44] C. Cheng and K. K. Parhi, "High-speed VLSI implementation of 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 393-403, Jan. 2008.
- [45] N. Usha Bhanu and A. Chilambuchelvan, "Efficient VLSI architecture for discrete wavelet transform," *Int. J. Computer Science Issues (Online)*, vol. 1, no. 1, pp. 32-36, Nov. 2011.
- [46] M. Ghantous and M. Bayoumi, "P²E-DWT: A parallel and pipelined efficient VLSI architecture of 2-D discrete wavelet transform," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, pp. 941-944, May 2011.
- [47] C. C. Liu, Y. H. Shiau, and J. M. Jou, "Design and implementation of a progressive image coding chip based on the lifted wavelet transform," *Proc. 11th VLSI Design/CAD Symp.* 2000.
- [48] J. M. Jou, Y. H. Shiau, and C. C. Liu, "Efficient VLSI architectures for the biorthogonal wavelet transform by filter bank and lifting scheme," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, pp. 529–532, May 2001.
- [49] C. J. Lian, K. F. Chen, H. H. Chen, and L. G. Chen, "Lifting based discrete wavelet transform architecture for JPEG2000," *Proc. IEEE Int. Symp. Circuits and Syst. (ISCAS)*, pp. 445–448, May 2001.
- [50] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 966–977, Apr. 2002.
- [51] C. T. Huang, P. C. Tseng, and L. G. Chen, "Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1080–1089, Apr. 2004.

- [52] W. H. Chang, Y. S. Lee, W. S. Peng, and C. Y. Lee, "A line-based, memory efficient and programmable architecture for 2D DWT using lifting scheme," *Proc. IEEE Int. Symp. Circuits and Syst. (ISCAS)*, pp. 330–333, May 2001.
- [53] H. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient implementation of lifting-based discrete wavelet transform," *Electronics Letters*, vol. 38, no. 18, pp. 1010–1012, Aug. 2002.
- [54] H. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient architectures for 1-D and 2-D lifting-based wavelet transform," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1315–1326, May 2004.
- [55] M. Martina, G. Masera, G. Piccinini, and M. Zamboni, "Novel JPEG 2000 compliant DWT and IWT VLSI implementations," *J. VLSI Signal Process. Systems Signal, Image Video Technology*, vol. 35, no. 2, pp. 137–153, Sep. 2003.
- [56] P. K. Meher, B. K. Mohanty, and M. N. S. Swamy, "Low-Area and Low-Power Reconfigurable Architecture for Convolution-Based 1-D DWT Using 9/7 and 5/3 Filters," *Proc. Int. Conf. VLSI Design*, pp. 327–332, Jan. 2015.
- [57] T. Acharya and C. Chakrabarti, "A survey on lifting-based discrete wavelet transform architectures," *J. VLSI Signal Process.*, vol. 42, no. 3, pp. 321–339, Mar. 2006.
- [58] C. Chakrabarti and M. Vishwanath, "Efficient realizations of the discrete and continuous wavelet transforms: From single chip implementations to mappings on SIMD array computers," *IEEE Trans. Signal Process.*, vol. 43, no. 3, pp. 759–771, Mar. 1995.
- [59] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," *IEEE Trans. Circuits Syst. II*, vol. 42, no. 5, pp. 305–316, May 1995.
- [60] C. Chakrabarti and C. Mumford, "Efficient realizations of analysis and synthesis filters based on the 2D discrete wavelet transform," *Proc. Int. Conf. Acoustics, Speech Signal Process.*, vol. 6, pp. 3256–3259, May 1996.
- [61] C. Chakrabarti, M. Vishwanath, and R. Owens, "Architectures for wavelet transforms: A survey," *J. VLSI Signal Processing*, vol. 14, no. 2, pp. 171–192, Nov. 1996.

- [62] C. Chrysafis and A. Ortega, "Line-based, reduced memory, wavelet image compression," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 378–389, Mar. 2000.
- [63] F. Marino, "Efficient high-speed/low-power pipelined architecture for the direct 2-D discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Analog Dig. Signal Process.*, vol. 47, no. 12, pp. 1476–1491, Dec. 2000.
- [64] P.-C. Tseng, C.-T. Huang, and L.-G. Chen, "Generic RAM-based architecture for two-dimensional discrete wavelet transform with line-based method," *Proc. Asia-Pacific Conf. Circuits Syst.*, vol. 1, pp. 363–366, Oct. 2002.
- [65] S. Barua, J. E. Carletta, K. A. Kotteri, and A. E. Bell, "An efficient architecture for lifting-based two-dimensional discrete wavelet transform," *Integr. VLSI J.*, vol. 38, no. 3, pp. 341–352, Jan. 2005.
- [66] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Generic RAM-based architectures for two-dimensional discrete wavelet transform with line-based method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 910–920, July 2005.
- [67] B. F. Wu and C. F. Lin, "A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1615–1628, Dec. 2005.
- [68] C.-Y. Xiong, J. Tian, and J. Liu, "Efficient high-speed/low-power line-based architecture for two-dimensional discrete wavelet transform using lifting scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 309–316, Feb. 2006.
- [69] C.-Y. Xiong, J.-W. Tian, and J. Liu, "Efficient architecture for 2-D discrete wavelet transform using lifting scheme," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 607–614, Mar. 2007.
- [70] B. K. Mohanty and P. K. Meher, "Memory efficient modular VLSI architecture for highthroughput and low-latency implementation of multilevel lifting 2-D DWT," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2072–2084, May 2011.
- [71] S. M. Aziz and D. M. Pham, "Efficient parallel architecture for multi-level forward discrete wavelet transform processors," *Comp. Elect. Eng.*, vol. 38, no. 5, pp. 1325–1335, Sep. 2012.

- [72] B. K. Mohanty and P. K. Meher “Memory-efficient high-speed convolution-based generic structure for multilevel 2-D DWT,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 353–363, Feb. 2013.
- [73] C.-H. Hsia, J.-S. Chiang, and J.-M. Guo, “Memory-efficient hardware architecture of 2-D dual-mode lifting-based discrete wavelet transform,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 4, pp. 671-683, Apr. 2013.
- [74] A. D. Darji, S. S. Kushwah, S. N. Merchant, and A. N. Chandorkar, “High-performance hardware architectures for multi-level lifting-based discrete wavelet transform,” *Eurasip J. Image Video Process.* vol. 47, pp. 1-19, Dec. 2014.
- [75] C.-H. Hsia, J.-S. Chiang, and S.-H. Chang, “An efficient VLSI architecture for 2-D dual-mode SMDWT,” *Proc. IEEE Int. Conf. Networking, Sensing and Control (ICNSC)*, pp. 775–779, Apr. 2013.
- [76] C.-H. Hsia, “A new VLSI architecture for symmetric mask-based discrete wavelet transform,” *J. Internet Technol.* vol. 15, no. 7, pp. 1083–1090, Dec. 2014.
- [77] V. Rajović, G. Savić, V. Čeperković, and M. Prokin, “Combined one-dimensional lowpass and highpass filters for subband transformer,” *Electronics Letters*, vol. 49, no. 18, pp. 1150-1152, Aug. 2013.
- [78] V. Rajović, G. Savić, and M. Prokin, “Hardware realization of fast image encoder with minimum memory size,” *Proc. 22nd Telecommunications Forum (TELFOR)*, pp. 717-724, Nov. 2014. (Invited paper)
- [79] Goran Savić, M. Prokin, V. Rajović, and D. Prokin, “Direktno filtriranje i dekompozicija slike sa minimalnim korišćenim resursima,” *Proc. 22nd Telecommunications Forum (TELFOR)*, pp. 725-728, Nov. 2014.
- [80] G. Savić, M. Prokin, V. Rajović, and D. Prokin, “Inverzno filtriranje i kompozicija slike sa minimalnim korišćenim resursima,” *Proc. 22nd Telecommunications Forum (TELFOR)*, pp. 729-732, Nov. 2014.
- [81] G. Savić, M. Prokin, V. Rajović, and D. Prokin, “Hardware realization of direct subband transformer with minimum used resources,” *Proc. 4th Mediterranean Conference on Embedded Computing (MECO)*, pp. 220-223, June 2015.
- [82] G. Savić, M. Prokin, V. Rajović, and D. Prokin, “Hardware Realization of Inverse Subband Transformer with Minimum Used Resources,” *Proc. 4th*

- Mediterranean Conference on Embedded Computing (MECO)*, pp. 224-227, June 2015.
- [83] G. Savić, M. Prokin, V. Rajović, and D. Prokin, “Optimalna hardverska realizacija direktnog podopsežnog transformatora,” *Tehnika*, vol. 64, no. 5, pp. 815-821, Oct. 2015.
- [84] G. Savić, M. Prokin, V. Rajović, and D. Prokin, “Optimal Hardware Realization of Direct Subband Transformer,” *Technics*, special edition, pp. 83-89, Dec. 2015.
- [85] V. Ceperkovic, S. Pavlović, D. Mirković, and M. Prokin, “Fast codec with high compression ratio and minimum required resources”, PCT/YU2003/000027.
- [86] G. Savić, M. Prokin, V. Rajović, and D. Prokin, “Novel one-dimensional and two-dimensional forward discrete wavelet transform 5/3 filter architectures for efficient hardware implementation,” *Journal of Real-Time Image Processing*, 2016. DOI: 10.1007/s11554-016-0656-1
- [87] G. Savić, M. Prokin, V. Rajović, and D. Prokin, “High-Performance 1-D and 2-D Inverse DWT 5/3 Filter Architectures for Efficient Hardware Implementation,” *Circuits, Systems & Signal Processing*, 2016. DOI: 10.1007/s00034-016-0477-2
- [88] G. Dillen, B. Georis, J. D. Legat, and O. Cantineau, “Combined line-based architecture for the 5–3 and 9–7 wavelet transform of JPEG2000,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 9, pp. 944–950, Sept. 2003.
- [89] X. Lan, N. Zheng, and Y. Liu, “Low-power and high-speed VLSI architecture for lifting-based forward and inverse wavelet transform,” *IEEE Trans. Consum. Electr.*, vol. 51, no. 2, pp. 379–385, May 2005.
- [90] L. Liu, N. Chen, H. Meng, L. Zhang, Z. Wang, and H. Chen, “A VLSI architecture of JPEG2000 encoder,” *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 2032–2040, Nov. 2004.
- [91] *DE2-115 User Manual*, ver. 2.1, Terasic Technologies, 2012.
- [92] *Cyclone IV Device Handbook – Volume 1*, ver. 1.8, Altera, February 2013.
- [93] *HSMC-DVI_User_Guide*, Terasic Technologies, 2010.
- [94] *Data Sheet Fujitsu Display B24W-6 LED*, 2013.
- [95] *Avalon Interface Specifications*, ver. 11.0, Altera, May 2011.

Biografija autora

Goran Savić je rođen u Sarajevu 23.11.1977. Osnovnu školu je završio 1992. godine u Sarajevu, a gimnaziju 1996. godine u Čačku. Elektrotehnički fakultet Univerziteta u Beogradu je upisao 1996. godine, gde je na Smeru za elektroniku diplomirao 2002. godine sa prosečnom ocenom 9,64 (najvišom u generaciji računajući sve odseke). Diplomski rad je odbranio na temu “Komunikacija sa električnim brojiлом preko IC porta po *CENELEC* standardu”. Na Elektrotehničkom fakultetu Univerziteta u Beogradu je magistrirao 2009. godine odbranivši magistarsku tezu pod nazivom “Hardverska realizacija brzog direktnog i inverznog filtra sa minimalnim resursima”.

Radni odnos na Katedri za elektroniku Elektrotehničkog fakulteta Univerziteta u Beogradu je zasnovao 19.05.2005. na određeno vreme u trajanju od četiri godine, sa punim radnim vremenom za obavljanje poslova u zvanju asistenta-pripravnika. U zvanje asistenta sa punim radnim vremenom na Katedri za elektroniku je izabran 07.09.2009, a u isto zvanje je reizabran 22.01.2013. Za vreme radnog angažmana je učestvovao u izvođenju računskih i laboratorijskih vežbi iz većeg broja predmeta. Od januara 2007. godine do januara 2016. godine je obavljao dužnost sekretara Katedre za elektroniku.

Koautor je priručnika za laboratorijske vežbe:

- M. Prokin, R. Đurić, J. Vujasinović, i G. Savić, “Osnovi elektronike (svi odseci osim odseka za elektroniku) – priručnik za laboratorijske vežbe,” Akademska misao, Beograd, 2006. (ISBN 86-7466-253-6)

Učestvovao je na sledećim projektima:

- Naučno-tehnološki projekat Ministarstva prosvete, nauke i tehnološkog razvoja Vlade Republike Srbije “Hardverska, softverska, telekomunikaciona i energetska optimizacija *IPTV* sistema,” TR32039, 2011-2017. (8 istraživač-meseci godišnje).

- Inovacioni projekat Ministarstva za nauku i tehnologiju Vlade Republike Srbije “Razvoj solarnih punjača izlazne snage do 10W za punjenje akumulatora,” 2007-2010. (8 istraživač-meseci godišnje).
- *Tempus JEP 17028-02*, projekat finansiran od strane Evropske komisije.

Iz oblasti doktorske disertacije, autor je ili koautor tri rada u istaknutim međunarodnim časopisima sa impakt faktorom sa *SCI* liste, pet radova na međunarodnim konferencijama i dva rada u domaćim časopisima, kao i dva rada na međunarodnim konferencijama van oblasti doktorske disertacije (označenih sa *):

Rad u istaknutom međunarodnom časopisu	M22
--	-----

- **Goran Savić**, Milan Prokin, Vladimir Rajović, and Dragana Prokin, “High-Performance 1-D and 2-D Inverse DWT 5/3 Filter Architectures for Efficient Hardware Implementation,” *Circuits, Systems & Signal Processing*, 2016. DOI: 10.1007/s00034-016-0477-2 (IF=1.178)
- **Goran Savić**, Milan Prokin, Vladimir Rajović, and Dragana Prokin, “Novel one-dimensional and two-dimensional forward discrete wavelet transform 5/3 filter architectures for efficient hardware implementation,” *Journal of Real-Time Image Processing*, 2016. DOI: 10.1007/s11554-016-0656-1 (IF=1.564)
- Vladimir Rajović, **Goran Savić**, Vladimir Čeperković, and Milan Prokin, “Combined one-dimensional lowpass and highpass filters for subband transformer,” *Electronics Letters*, vol. 49, no. 18, pp. 1150-1152, Aug. 2013. (IF=1.038)

Saopštenje sa međunarodnog skupa štampano u celini	M33
--	-----

- **Goran Savić**, Milan Prokin, Vladimir Rajović, and Dragana Prokin, “Hardware Realization of Direct Subband Transformer with Minimum Used Resources,” *Proceedings of 4th Mediterranean Conference on Embedded Computing (MECO) 2015*, Budva, Montenegro, 14-18 June 2015, pp. 220-223.

- **Goran Savić**, Milan Prokin, Vladimir Rajović, and Dragana Prokin, “Hardware Realization of Inverse Subband Transformer with Minimum Used Resources,” *Proceedings of 4th Mediterranean Conference on Embedded Computing (MECO) 2015*, Budva, Montenegro, 14-18 June 2015, pp. 224-227.
- Vladimir Rajović, **Goran Savić**, and Milan Prokin, “Hardware Realization of Fast Image Encoder with Minimum Memory Size,” *Proceeding of 22nd Telecommunications Forum (TELFOR) 2014*, Belgrade, Serbia, 25-27 Nov. 2014, pp. 717-724.
- **Goran Savić**, Milan Prokin, Vladimir Rajović, and Dragana Prokin, “Direktno filtriranje i dekompozicija slike sa minimalnim korišćenim resursima,” *Proceedings of 22nd Telecommunications Forum (TELFOR) 2014*, Belgrade, Serbia, 25-27 Nov. 2014, pp. 725-728.
- **Goran Savić**, Milan Prokin, Vladimir Rajović, and Dragana Prokin, “Inverzno filtriranje i kompozicija slike sa minimalnim korišćenim resursima,” *Proceedings of 22nd Telecommunications Forum (TELFOR) 2014*, 25-27 Nov. 2014, Belgrade, Serbia, pp. 729-732.
- D. Prokin, and **G. Savić**, “Fastest Lowest Complexity Bit-Pipeline Rank Filter for FPGA Implementation,” *Proceedings of International Scientific Conference UNITECH*, Gabrovo, Bulgaria, 18-19 Nov. 2011, pp. I-235 - I-240. (*)
- S. Marjanović, J. Vujasinović, **G. Savić**, S. Manojlović, B. Ninković, and P. Međedović, “Sistem za daljinsko očitavanje i upravljanje potrošnjom električne energije,” *Zbornik radova VI Savetovanja Bosanskohercegovačkog Komiteta CIGRE*, Neum; Bosna and Hercegovina, Nov. 2003, vol. RC6-02, pp. 1-5. (*)

Rad u časopisu nacionalnog značaja	M52
------------------------------------	-----

- **Goran Savić**, Milan Prokin, Vladimir Rajović, i Dragana Prokin, “Optimalna hardverska realizacija direktnog podopsežnog transformatora,” *Tehnika*, vol. 64, no. 5, 2015, pp. 815-821, ISSN 0040-2176
- **Goran Savić**, Milan Prokin, Vladimir Rajović, and Dragana Prokin, “Optimal Hardware Realization of Direct Subband Transformer,” *Technics, special edition*, 2015, pp. 83-89, ISSN 0040-2176

Прилог 1.

Изјава о ауторству

Потписани-а Горан С. Савић

Број индекса _____

Изјављујем

да је докторска дисертација под насловом

Хардверска реализација брзог декодера компримоване слике са минималним
захтеваним ресурсима

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис докторанда

У Београду, 01.03.2017. године

Савић Горан

Прилог 2.

**Изјава о истоветности штампане и електронске
верзије докторског рада**

Име и презиме аутора Горан С. Савић

Број индекса _____

Студијски програм Електротехника и рачунарство

Наслов рада Хардверска реализација брзог декодера компримоване слике са
минималним захтеваним ресурсима

Ментор др Милан Прокин, редовни професор

Потписани Горан С. Савић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањена у **Дигиталном репозиторијуму Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 01.03.2017. године

Савић Горан

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Хардверска реализација брзог декодера компримоване слике са
минималним захтеваним ресурсима

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство – некомерцијално

3. Ауторство – некомерцијално – без прерада

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прерада

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци.

Кратак опис лиценци је саставни део ове изјаве).

Потпис докторанда

У Београду, 01.03.2017. године

Calvin Jayant

1. **Ауторство.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.